THE Magazine for experienced TANDY Colour Computer Users!

$4.50

AUSTRALIAN

COCO MAGAZINE

VOL4NO2
OCT87

Registered by Australia Post: Reg. No. QBG 4007 ISSN 0819-9035

BLAXLAND COMPUTER SERVICES PTY. LTD.

SERVICING MODERN TECHNOLOGY

Tandy ELECTRONICS DEALER

NO 9254

SOFTWARE-SOFTWARE-SOFTWARE-SOFTWARE-SOFTWARE-SOFTWARE-SOFTWARE-SOFTWARE

OS-9 LEVEL 2 - Includes Basic 09 (Direct from U.S.)	CC3 - $180.00
DESKMATE 3 - For your Coco 3	CC3 - P.O.A.
I.M.S.- 4th Generation Language Database (Single User)	CC2/3 - $299.00
SCULPTOR - 4GL Database Multiuser (OS-9 level 2 req.)	CC3 - $1200.00
THE WIZ - Communications with Windows (L2 & RS232 req.)	CC3 - $160.00
SCREEN STAR - OS-9 Screen Editor with Smart Speller	CC3 - $99.95
SCREEN STAR TEXT FORMATTER - use with Screen Star	CC3 - $69.95
RAMDISK & 512k DIAGNOSTIC - plus utilities	CC3 - $39.95
COLOR SCRIBE - RS/DOS Wordprocessor	CC3 - $99.95
THE WILD WEST - by Tom Mix (Disk)	CC3 - $51.95
NUKE THE LOVE BOAT - Adventure using 512k as ramdisk	CC3 - $69.95
MAGIC OF ZANTH - Adventure with top graphics	CC3 - $69.95
RETURN OF JUNIOR'S REVENGE - All time favourite	CC3 - $69.95
PRO GOLF - 36 Holes, 32K required (Disk)	CC2 - $59.95
MR. DIG - Tape or Disk	CC2 - P.O.A.
ESCAPE: 2012 - 64K Adventure (137 rooms)	CC2 - P.O.A.
TREASURE OF THE AZTECS - 4 voice music, 50 hires screens	CC2 - P.O.A.
ROBOT ODYSSEY - 64K Adventure (Disk)	CC2 - P.O.A.
COMPLETE RAINBOW GUIDE TO OS-9 - 2 Disks, no book	CC2 - $59.95
FIRST RAINBOW BOOK OF ADVENTURES - Book and Tape	CC2 - $32.00
SECOND RAINBOW BOOK OF ADVENTURES - Book and Tape	CC2 - $56.00
THIRD RAINBOW BOOK OF ADVENTURES - Book and Tape	CC2 - $44.00
" " " " " - Book and Disk	CC2 - $54.00
RAINBOW BOOK OF SIMULATIONS - Book and Tape	CC2 - $40.00
SECOND RAINBOW BOOK OF SIMULATIONS - Book and Tape	CC2 - $40.00
INTRODUCTORY GUIDE TO STATISTICS - Book and Tape or Disk	CC2 - $26.00
DISK TUTORIAL - for serious disk basic/ml prog. (2 disk)	CC2/3 - $74.00
COCO GRAPHICS DESIGNER - print your cards, signs, banners	CC2/3 - $59.95
" " " - 100 pre-drawn pictures	CC2/3 - $29.95
3D-GRAPHIMATOR - 3 Dimensional Drawing Design	CC2 - P.O.A.
PROGRAMMERS UTILITY - Commands & Utilities by keystroke	CC2 - P.O.A.
PRINTMASTER	IBM - $96.35
ART GALLERY 1	IBM - $56.65
ART GALLERY 2	IBM - $56.65
NEWSROOM	IBM - $101.95
CLIP ART 1	IBM - $50.95
CLIP ART 2	IBM - $62.35
CLIP ART 3	IBM - $62.35
NEWSMASTER (DESKTOP PUBLISHING)	IBM - $170.00
PRODESIGN (CAD SYSTEM WITH HI-RES PRINTOUTS)	IBM - $630.00
CLIPPER (DBASE III COMPILER)	IBM - $599.00
DAC-EASY ACCOUNTING AUSTRALIAN VERSION	IBM - $250.00
DAC-EASY WORD PROCESSOR	IBM - $129.00
FBS ACCOUNTING SUITE (Inventory,Debtors,Creditors,G/L)	IBM - $830.00
FBS PAYROLL	IBM - $630.00
FBS BILL OF MATERIALS	IBM - $630.00
FBS JOB COSTING	IBM - $630.00
AI TYPIST WORD PROCESSOR	IBM - $159.00
PEEKS N' POKES FOR IBM	IBM - $66.00
INSIDE TRACK FOR IBM	IBM - $85.00
SPEEDIT PROGRAM EDITOR	IBM - $79.00
PROKEY	IBM - $195.00
MUSIC+ II DISK OR TAPE	CC2 - $49.00
MUSICA LIBRARY 100/800 EACH SET	CC2 - $49.00
GEOGRAPHY (Africa,South America,Asia,Europe,U.S.A.)	CC2 - $99.95
BOUNCING BOULDERS	CC2/3 - $49.95
GAUNTLET	CC2/3 - $49.95
KARATE	CC2/3 - $49.95
KNOCKOUT	CC2/3 - $49.95
F-16 MISSION ASSAULT	CC2/3 - $49.95
SHOCK TROOPER	CC2 - $33.45
PAPER CHASE	CC2/3 - $49.95
MARBLE MAZE	CC2/3 - $49.95
DECATHLON	CC2 - $33.45
MONEYPOLY	CC2 - $33.45
SAILOR MAN	CC2 - $33.45
COCOTEX VIATEL SOFTWARE	CC2/3 - $99.95
BUSINESS ACCOUNTING SYSTEM	CC2 - $99.95
FAMILY TREE GENEOLOGY	CC2 - $33.45
SUPER BACKUP UTILITY	CC2 - $59.95
TELEWRITER 64 DISK + TAPE	CC2 - $84.45
VIP-WRITER DISK + TAPE	CC2 - $98.95
VIP-DATABASE DISK ONLY	CC2 - $84.45
VIP-CALC DISK + TAPE	CC2 - $84.45
VIP-SPELLER DISK ONLY	CC2 - $77.45
VIP-TERMINAL DISK + TAPE	CC2 - $84.45

BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS

AMERICAN RAINBOW - Direct from USA, month of publication	P.O.A.
BACK ISSUE ORDERS - place your order in July and August	P.O.A.
P.C.M. - Portable Computer Monthly, month of publication	P.O.A.
INSIDE OS-9 LEVEL II - A must for Level II	$79.00
500 PEEKS and POKES - for your Coco	$33.95
200 ADDITIONAL PEEKS and POKES - supplement to 500	$19.95
ASSEMBLY LANGUAGE PROGRAMMING - by Barden	$11.95
COMPLETE RAINBOW GUIDE TO OS-9	$39.95
NANOS REFERENCE CARD FOR COCO	$5.95
6809 ASSEMBLY LANGUAGE PROGRAMMING	$41.95
VISICALC APPLICATIONS	$19.95
MULTIPLAN APPLICATIONS	$19.95
UNDERSTANDING COMPUTER SCIENCE	$5.49
DBASE III TIPS AND TRAPS	$37.95
GRAPHICS PRIMER FOR YOUR IBM PC	$45.95
YOUR IBM MADE EASY	$31.99
PC SECRETS	$35.95
PC DOS TIPS & TRAPS	$35.95
YOUR IBM PC	$39.95
GUIDE TO USING LOTUS 1-2-3	$36.95
C LIBRARY	$36.95
USING DBASE III	$37.95
THE 8086 BOOK	$39.95
THE SHAREWARE BOOK	$31.95
MBASIC HANDBOOK	$39.95

HARDWARE-HARDWARE-HARDWARE-HARDWARE-HARDWARE-HARDWARE-HARDWARE-HARDWARE

10MEGABYTE HARD DISK DRIVE - Speed and Mass Storage	CC2/3 - $1299.00
20MEGABYTE HARD DISK DRIVE - Large Business Storage	CC2/3 - $1599.00
512k COCO 3 UPGRADE Includes P.D Software	CC3 - $220.00
TEAC DSDD DRIVE 40T PLUS CONTROLLER - Double Height	CC2/3 - $680.00
TEACx2 DSDD DRIVES 40T PLUS CONTR. - Equal to 4 Tandy	CC2/3 - $950.00
BARE TEAC DSDD 40T DRIVES - Hi-tech reliability	CC2/3 - $300.00
BARE TEAC DSDD 80T DRIVES - 1.4MB Storage per drive	CC2/3 - $350.00
BARE TANDY SSDD DRIVES - not $519 (subject to stock)	CC2/3 - $250.00
T1000 20MB INTERNAL HD - 1 year warranty, formatted	T1000 - $1399.00
T1000 COMBI RS232/640K/CLOCK/CALENDAR	T1000 - $750.00
THOMSON AMBER/GREEN COMPOSITE MONITORS WITH SOUND	ALL - $495.00
TANDY 1000 SX - 384K, 2 DRIVE	T1000 - $1999.00
TANDY 1000 EX	T1000 - $1299.00
TANDY COLOUR COMPUTER 3 - 128K	CC3 - $449.00
TANDY 102 PORTABLE	T102 - $999.00
TANDY 200 PORTABLE	T200 - $1399.00
MULTIPAK INTERFACE - SWITCHABLE COCO 2/3, LED (ON/OFF	CC2/3 - $199.95
VIDEO DRIVER WITH SOUND	CC2 - $36.00
VIDEO DRIVER WITHOUT SOUND (if your monitor has sound)	CC2 - $32.00
MODIFIED ARCHER JOYSTICKS - SUIT COCO 2/3	CC2/3 - $27.00
CITIZEN 120D DOT MATRIX PRINTER - EXTRA TYPEFONTS	ALL - $599.00
DMP-106 DOT MATRIX PRINTER	ALL - $399.00
DMP-130 DOT MATRIX PRINTER	ALL - $599.00
AVTEK MINI MODEM - 300 & 1200/75 baud, economy plus	ALL - $260.00
TANDY INTERNAL MODEM - 300 & 1200/75 + VIS32 software	T1000 - $699.00
SUPER AUTO MODEM - 300,1200/75,1200 auto ans./dial/con.	T1000 - $820.00
WYSE 30 TERMINALS	CC3 - $999.00

-MISCELLANEOUS-MISCELLANEOUS-MISCELLANEOUS-MISCELLANEOUS-MISCELLANEOUS-

PRINTER PAPER 9-1/16" x 11" PLAIN per 1000 sheets	$17.00
PRINTER PAPER " x " PLAIN per box 4000 sheets	$66.00
PRINTER PAPER 9-1/2" x 11" PLAIN per 1000 sheets clean edge	$24.00
PRINTER PAPER 9-1/2" x 11-2/3" PLAIN A4 per 1000 sheets	$29.00
PRINTER PAPER 11" x 15" PLAIN per 1000	$30.00
PRINTER PAPER 11" x 15" RULE RULE per 1000 (also SCREEN)	$27.00
NCR 2 PART 9-1/2" x 11" (no carbon required) per 1000	$74.00
MAILING LABELS 11mm x 54mm (4 up) per 1000	$6.00
MAILING LABELS 23mm x 89mm (1 up) per 1000	$16.00
MAILING LABELS 36mm x 102mm (1 up) per 1000	$16.00
CASSETTE LABELS (orange) per100 (10 per sheet $5.00 (white)	$4.00
BLANK CASSETTES C-10's (10 min.) $1.10 each. C-90's	$1.80
STORAGE CASE, HINGED 5-1/4" holds 100 $25: holds 60	$22.00
STORAGE CASE, HINGED 3-1/2" holds 60	$15.00
3-1/2" DSDD DISKETTES box 10	$55.00
5-1/4" DSDD DISKETTES box 10	$25.00
DISK HEAD CLEANING KIT 3-1/2"	$11.95
DISK HEAD CLEANING KIT 5-1/4"	$11.95
LARGE RANGE PRINTER RIBBONS/CARTRIDGES (Tandy/non Tandy)	P.O.A.
LARGE RANGE PRE-PRINTED INVOICE/STATEMENT FORMS AVAILABLE	P.O.A.

bankcard MasterCard

VISA

AGC CREDIT LINE

BLAXLAND COMPUTER SERVICES PTY. LTD.

134 GREAT WESTERN HIGHWAY,
BLAXLAND 2774 N.S.W.
P.O BOX 125
PHONE: (047) 39 3903

PRICES SUBJECT TO CHANGE WITHOUT NOTICE.

POSTAGE AND PACKAGING EXTRA

the GOLDSOFT WISHBOOK

The Goldsoft Wishbook

The following products are available on order from us.

To order, contact us by phone, Viatel or letter, giving your name, address, phone number and credit card number, as well as the Item # shown beside the product as listed below.

All items include post and packing.

CoCo Hardware

Item #	Description	Price
G 001	The CoCoConnection - Use your CoCo to control models,alarms - anything electrical	$206.00
G 002	Video Amplifier with sound - attach your CoCo 1 or 2 to a Video monitor	$35.00
G 003	The Probe - A temperature sensing unit you plugin to the joy stick port.	$49.95

CoCo Software

Item #	Description	Price
G 1001	Say the Wordz - two Curriculum based speller programs for your Tandy Speech / Sound Pack (32K ECB)	$29.95

The CoCo 3 Tape/Disk

Item #			Price
G 1002		# 1	$16.00
G 1003		# 2	$16.00
G 1004		# 3	$16.00
G 1015	#10	Education	$16.00
G 1016	#11	Education (Disk only)	$16.00

The Best of CoCoOz

Item #	Description	Price
G 1005	# 1 Education	$16.00
G 1006	# 2 Part 1 16K Games	$16.00
G 1007	# 2 Part 2 32K Games	$16.00
G 1008	# 3 Utilities	$16.00
G 1009	# 4 Business	$16.00
G 1010	# 5 Adventure Games	$16.00
G 1011	# 6 Preschool Edn	$16.00
G 1012	# 7 Graphics	$16.00
G 1013	# 8 16K Games	$16.00
G 1014	# 9 32K Games	$16.00

Tandy and IBM PC Hardware

Item #	Description	Price
Q 1006	Crosstalk	$306.00
Q 1007	Lotus 123	$1054.00
Q 1008	Wordstar 2000+	$927.00
Z 2001	Webster's New World Writer	$249.00
Z 2002	Webster's Spelling Checker	$89.95
Z 2003	Webster's Thesaurus	$89.95
Z 2004	Windowword	$269.00
Z 2005	Ready	$99.00
Z 2006	Thinktank	$385.00

CAD Packages

Item #	Description	Price
Z 2007	TURBOCAD (V 1.4)	$399.00

Trade Business Packages

Item #	Description	Price
Z 2008	The Motor Trade Package	$99.00
Z 2009	The Professional's Pack	$99.00
Z 2010	The Retailer's Package	$99.00
Z 2011	The Rental Package	$99.00

Accounting

Item #	Description	Price
Z 2012	Asset Manager	$1170.00
Z 2013	Cash Desk/Finance Desk	$399.00
Z 2014	System 4	$645.00
Z 2015	C.P.A. Plus	$395.00

Databases

Item #	Description	Price
Z 2016	Omnus3 IBM-Single user (Multi user versions are available)	$495.00

Spreadsheets

Item #	Description	Price
Z 2017	Logistix	$399.00

Fun

Item #	Description	Price
Q 1020	Ancient Art of War	$96.00
Q 1021	Print Shop	$119.00
Q 1022	Gato	$68.00
Q 1023	Sargon III	$96.00
Q 1024	Zork I	$79.00
Q 1025	Zork II	$79.00
Q 1026	Zork III	$79.00
Q 1027	Trinity	$79.00
Q 1028	Ballyhoo	$79.00
Q 1029	Hitch Hicker's Guide to the Galaxy	$79.00
Q 1030	Crossword Magic	$68.00
Q 1031	The American Challenge	$68.00
Q 1032	Balance of Power	$89.00
Q 1033	Racter	$79.00
Q 1034	Jet	$114.00
Q 1035	Moonmist	$79.00
Q 1036	Shanghai	$68.00
Q 1037	Championship Golf	$89.00
Q 1038	Borrowed Time	$68.00
Z 2018	The Great International Paper Airplane Construction kit	$49.95
Z 2019	Star Trek	$49.95
Z 2020	Championship Boxing	$69.95
Z 2021	Ultima II	$69.95
Z 2022	Decision in the Desert	$69.95
Z 2023	F-15 Strike Eagle	$69.95
Z 2024	Kings Quest	$69.95
Z 2025	Mean 18	$69.95
Z 2026	Boulderdash	$49.95
Z 2027	Boulderdash II	$49.95
Z 2028	Conflict in Vietnam	$69.95
Z 2029	Dambusters	$69.95
Z 2030	Kings Quest II	$69.95
Z 2031	PSI-5 Trading Company	$69.95
Z 2032	Silent Service	$69.95
Z 2033	Solo Flight	$69.95
Z 2035	Star Fleet	$59.95

Education

Item #	Description	Price
Z 2036	Chem Lab	$69.95
Z 2037	Creature Creator	$59.95
Z 2038	Crypto Cube	$59.95
Z 2039	Decimal Dungeon	$49.95
Z 2040	Donald Duck's Playground	$59.95
Z 2041	European Nations and Locations	$59.95
Z 2042	Fraction Action	$49.95
Z 2043	Math Maze	$59.95
Z 2044	Mickey's Space Adventure	$69.95
Z 2045	Mission Algebra	$59.95
Z 2046	Race Car 'Rithmetic	$49.95
Z 2047	Remember!	$89.95
Z 2048	Ships Ahoy	$59.95
Z 2049	Spellagraph	$59.95
Z 2050	Spellakazam	$59.95
Z 2051	Spellicopter	$59.95
Z 2052	Ten Little Robots	$49.95
Z 2053	Winnie The Pooh in the 100 Acre Wood	$69.95

Miscellaneous Items

Item #	Description	Price
G 2001	Box of 10 DSDD Disks	$19.00
	10 Boxes plus (per box)	$16.10
G 2002	Blank C30 Cassettes	$2.00
	12 Cassettes	$18.00
G 2003	Tape cases, 12 for	$5.00
G 2004	Help - Manual for CoCo	$9.95

Terminal Programs

Item #	Description	Price
G 1017	CoCoTex - Videotex pac (Viatel) for all CoCos On Tape	$79.95
G 1018	CoCoTex as above on disk	$79.95
G 1019	Vtex 2 - Videotex pac for IBM Compatibles	$225.00
G 1020	Supertex 2 for Amiga & Atari 520 ST (specify)	$99.95
G 1021	Interlink - Videotex software for IBM PC's and compatibles	$94.95

Modems/Software/Cable

Item #	Description	Price
G 005	CoCoTex with cable and manual modem (Specify disk or tape)	$295.00
G 006	CoCoTex with cable and auto dial modem (Specify disk or tape)	$451.00
G 007	Interlink with cable and manual modem	$385.00
G 008	Vtex 2 with half card auto modem	$555.00
G 009	Vtex 2 with desktop auto modem and cable	$720.00

Other Hardware

Item #	Description	Price
A 001	Gender Changer - Male to Male	$12.45
A 002	Gender Changer Female to Female	$12.75
A 003	RS232 Data Switch with Tester	$99.95

inside COCO

Utilities

Articles

Games

Music

Graphics

Education

Founder: Greg Wilson.
Managing Editor: Graham Morphett.
Editor: Alex Hartman.
Accounts: Karen Court
Production: Sheryl Bentick
Advertising: Graham Morphett.
Sub Editors:
OS9: Jack Fricker
FORTH: John Redmond

Special Thanks to:
Fred Bisseling, Arthur Slade
Johanna Vagg, Martha Gritwhistle.
Deadlines: 7th of preceeding month.
Phone: 075 396177 or contact on Viatel *64213#
Registered Publication QBG 4009.

IN A NUT SHELL

What a month! Just when I thought there was nothing new, here were all these changes! A few of these were:

* Deon & Meredith

Deon is our new Tandy 1000 person. So if you have a submission to make to the Tandy 1000 section of Softgold magazine, send it to Deon.

If you submit an article or program or whatever, the same conditions apply when you send in submissions to the CoCo section of either CoCo or Softgold - Read the article on "Submitting your Work".

Also, if you have a question regarding earlier Tandy machines such as the Model 1, 3 or 4 or anything to do with MS-DOS equipment, ask Deon. He has a solution to (almost) everything.

Meredith is our new paste-up person. So, if you see any nasty boo-boos in the magazine (which is unlikely, for she's VERY good at what she does), you know that Meredith's been at it - naturally, she's still learning.

This magazine is her first effort - not bad, either.

* Goldlink *642#

"Who?" - that's the reaction we got when we finally got back on-line.

We were off-line for two weeks or so during September due to a dispute with Viatel and our billing. The problem is still unresolved, but hopefully we can finalize it without going off-line again.

Now that we're back, let's party!

* New "Best of" Tapes

In about a month or two, we will be releasing an all new "Best of" series of tape and disks. The subjects on each one will be:

- #12: 16K Games III
- #13: 32K Games III
- #14: CoCo 3 games
- #15: 16K Utilities
- #16: 32K & 64K Utilities
- #17: CoCo 3 Utilities
- #18: CoCo 2 & 3 Graphics
- #19: Education IV
- #20: Adventures II

Most of them are in the development stage and they reflect excellent quality of the magazine - so go to 'em (when we officially announce their release!).

Apart from all the changes, here are some other subjects worth mentioning.

* January Reference Issue

As you may know, the January edition of CoCo, is set aside as the reference issue. In that month, we print information such as error messages, computer vocabulary, computer terms, "peeks, pokes 'n execs" and so on.

We are looking NOW for anything that might be useful for such an edition.

It might be the printer codes for your DMP-200 printer (- we hope to print all the printer codes for each printer so you can easily convert that !#$@@! printer program that won't work on your DMP-130 to something that will work on your printer), or an article on OS-9 terms, or general hints and tips, or ... or ... anything you like!

But we have to have your information early by the the 7th of November, 1987 so we can put it in January's issue.

Please note that we make January's edition late November.

In the last two weeks of December, we have an official holiday (yes, we have holidays, believe it or not!).

* 64K Upgrade Kits

Those who wanted the 64K upgrade kits - terribly sorry for all the delay. We can't seem to deliver what we promised, because of the 'little' trade war between America and Japan!

We hope that we deliver within the next month.

* Surveys

Remember that infernal question and answer sheet we published in September's magazine - the survey form? Well, because we've had such a wonderful response, we are going to extend the deadline to the 21st of October.

We're receiving we'd really like to collate the surveys and publish a report in the following months magazine.

The competition to win 100 disks has been extended to the 21st as well!

See ya!

Alex

LETTERS

Dear Graham & Alex,

I read letters and articles printed in your magazines and feel a lot of the information is worth its weight in gold.

I would like to contribute a little, hopefully useful information.

In your last two magazines you have had people wanting information regarding screen dumps onto a Brother printer.

I have a friend who has a CoCo 3 with a Brother printer. He, too, was pulling out his hair because he couldn't get screen dumps. At his age, he could not afford to pull out any hair - he's seventy years young. Age is no barrier when learning on computers.

Firstly with the Brother printer, you must set the dip switches for either graphics or text.

The settings for text are as follows:

Dip Switch 1: 1-off, 2 & 3 & 4 & 5-on, 6-off, 7 & 8-on, 9-off, 10-on.

Dip switch 2: 1 & 2-on, 3 & 4 & 5 & 6-on, 7 & 8 & 9-off, 10-off.

The settings for graphics are the same for dip switch 1, but for dip switch 2, they are: 1-on, 2-off, 3-on, 4-off, 5-on, 6-off, 7 & 8-on, 9 & 10-off

Now for the screen dump program, parts of which come from many late nights.

```
1 SCREEN1,1 'DISPLAYS PICTURE
8 DIM D(15):FOR A=0 TO 15:READ D(A):NEXT
10 PRINT#-2,CHR$(27)"@"CHR$(27)"A"CHR$(8)CHR$(27)"2":S$=CHR$(27)+"K"+CHR$(128)+CHR$(1):FOR A=1536 TO 1567:FOR X=1 TO 2:PRINT#-2,S$;:FOR B=191 TO 0 STEP -1:P=NOT(PEEK(A+32*B)):IF X=1 THEN C=(P AND 240)/16 ELSE C=P AND15
15 PRINT#-2,CHR$(D(C));CHR$(D(C));:NEXT B:PRINT#-2,CHR$(10):NEXT X,A:PRINT"OKAY, FINISHED."
20 DATA0,3,12,15,48,51,60,63,192,195,204,207,240,243,252,255
```

Please note that this program does a print out of your screen from left to right going down the page. The next bit of information is for a password on the CoCo 3 when using the WIDTH 32 column screen.

Just insert the following program somewhere at the start of your program.

```
1 WIDTH40:PALETTE13,63:PALETTE13,64:PRINT
2 INPUT"PASSWORD PLEASE";A$'PUT ANY PASSWORD PROGRAM HERE
3 'IF PASSWORD IS INCORRECT
4 WIDTH32
5 'IF PASSWORD IS CORRECT
6 'START PROGRAM
```

Going back to graphics to get a copy of a PMODE 4 screen onto disk, or just the picture on the screen, press <reset> or <break> and type in ...

```
SAVEM"filename",3584,9727,3584
```
(for disk)

```
CSAVEM"filename",1536,7679,1536
```
(for tape)

To load either type in ...

```
PMODE4:PCLS:SCREEN1,1
(C)LOADM"filename" <enter>
```

To view the screen, type ...

```
SCREEN1,1:EXEC44539
```

Last but not least, we were all beginners at one stage. I notice in your "Help" section a lot of people write in about programs that have errors such as ?FC or ?TM.

To fix this, people must sit down and study the program - they will find the line causing the error is not in that line, but in some other line.

Eg, 150 A$=K$:B$=S$(5)

There is no error in line 150, so you get complaints. All they have to do is to find the value of A$ or whatever is causing the error by asking the computer to PRINT A$ or whatever.

I hope these hints will come in handy to users.

If anyone wants to contact me regards more Brother printer information or just for a natter about my dear old grey case (computer).

Phone 086-362-546 anytime, just ask for Paul.

Paul Savage,
Crystal Brook, SA

Paul,

Thank you for including your telephone number.

One of the things that is coming out in the surveys is that many people would like to write to authors - or at least be able to phone them. But as it is
a Goldsoft policy to protect authors' privacy, we are only prepared to reveal addresses and telephone numbers where the author specifically includes this information in line 1 of this program, or where it is included in the article which describes this program.

Graham

Hints and Tips

RENUM blues

Ever lost your BAsic program due to a RENUM command, ie you find that your Basic program has been lost when you've RENUMbered it?

Yes? Then type this in - this'll restore that program for you.

```
EXEC &HACEF:CLEAR
```

*

ECB system - CoCo 3 only

The next group of commands create an ECB system. Particularly useful when loading programs off tape that require an ECB environment, without having to unplug the controller.

```
POKE 49152,0:POKE&H134,0:POKE&H139,0:EXEC&H8002
```

NOTE: POKE49152,68:EXEC&H8002 will restore everything to normal.

Black and White Screens (CoCo 3)

Wanted a screen with a (colour) difference? Below are the PALETTES to use if you want a white on black screen.

Width 32:

```
PALETTE12,63:PALETTE13,0:CLS
```

Width 40/80 mode:

```
PALETTE0,0:PALETTE8,63:CLS1
```

REVIEW

by Arthur Slade

Software: Deskmate 3

DESKMATE 3 IS specifically designed for the CoCo 3. It comes with 2 Disks. The first disk contains the operating systems and the Ledger program, where as the second disk has the remainder of the programs, e.g. Calendar, Text, Filer as well as Paint and Telecom.

The disks, on a 2 drive system, switch between each other depending on the application you are using.

However, you are prompted when to change disks, should you only have one drive.

A very comprehensive 205pg manual and a reference card as well come with the package, making it MOST easy to understand.

It requires DECB 2.1 and all you do is type 'DOS' and <enter>.

When Deskmate is booted you are requested (if you want) to put in the date and time.

As the program is written under OS-9 Level 2, you can therefore boot your OS-9 level 2 disk, and at the OS9 prompt type ...

```
OS9:RENAME /D0/CMDS/AUTOEX
 DESKMATE <enter>
```

This changes the AUTOEX file to DESKMATE. At this point there are 2 ways you can access Deskmate programs; from the command line type ...

```
OS9:DESKMATE      <enter>,   or
OS9:DESKMATE DMxxxx  <enter>
```

... where 'DMxxxx' are any of the deskmate program filenames.

For example, say you only wanted TEXT EDITOR and the sample you typed in. At the prompt, type ..

```
OS9:DESK DMTEXT1:SAMPLE  <enter>
```

However all of these features are discussed fully in the manual. There are 8 utilities you can use, acessing each of them by pressing 'ALT*'.

These eight utilities are ...

* FOLDER,
* PRINTER,
* CALCULATOR,
* TIME,
* DISPLAY,
* CURSOR CONTROL,
* HELP and
* ICON.

I will give a brief description of 'Display'.

This utility is designed to let you change the resolution on screen (in TELECOM-TEXT-LEDGER) to a 40 or 80 column format and change the 16 available colours on the colour palette.

The calendar and paint files can only use 40 columns.

The colour palette consists of 3 scales: red, green and blue with which you can alter any of the available colours.

The top 4 colours are specifically designed for the screens; background (BG), foreground (FG), window borders (WB) and the command bar (CB).

Paint is autoset for 16 colour format.

ALWAYS REMEMBER TO BACKUP YOUR DISKS. This can be done firstly by write-protecting your originals, booting Deskmate, and using Deskmate's own FORMAT and BACKUP facilities that are provided.

REMEMBER: You don't have to have two drives to use Deskmate.

You have the option of either Mouse/Joystick or Keyboard.

Much can be written in a review about this particular program, however I can only give you my opinion, and after having spent the best part of a week playing around with it on my 512K CoCo 3, I could find no fault whatsoever.

In fact with the added HELP menus, it is a very user friendly program, and after a couple of days I found I only needed to refer to the reference card, which is double sided.

If you find you need to set up a budget, create a mailing list, set up a calendar schedule or even DRAW a picture, write a letter, then THIS program is for YOU!

I fully recommend it to those of you who own a CoCo 3 - you have everything at your finger-tips.

For a price of $129.95 it IS value, and I rate it a 10/10.

Review product gratefully supplied by Tandy (INTERTAN).

Tandy Products, 91 Kurrajong Avenue, Mt.Druitt. N.S.W. 2770.

Cat No.26-3262.

Look for it at your local Tandy store.

Hints and Tips

*

COPY with fewer swaps

The following will copy fairly large programs on disk with fewer swaps.

```
CLEAR200,32766:POKE25,14:POKE358
4,0:NEW
```

*

*

INKEY redefined

Waiting for a keystroke? Then do it this way! Instead of:

```
10 A$=INKEY$:IFA$=""THEN10
```

... try:

```
10 EXEC44539:A$=INKEY$
```

*

FOLK MUSIC

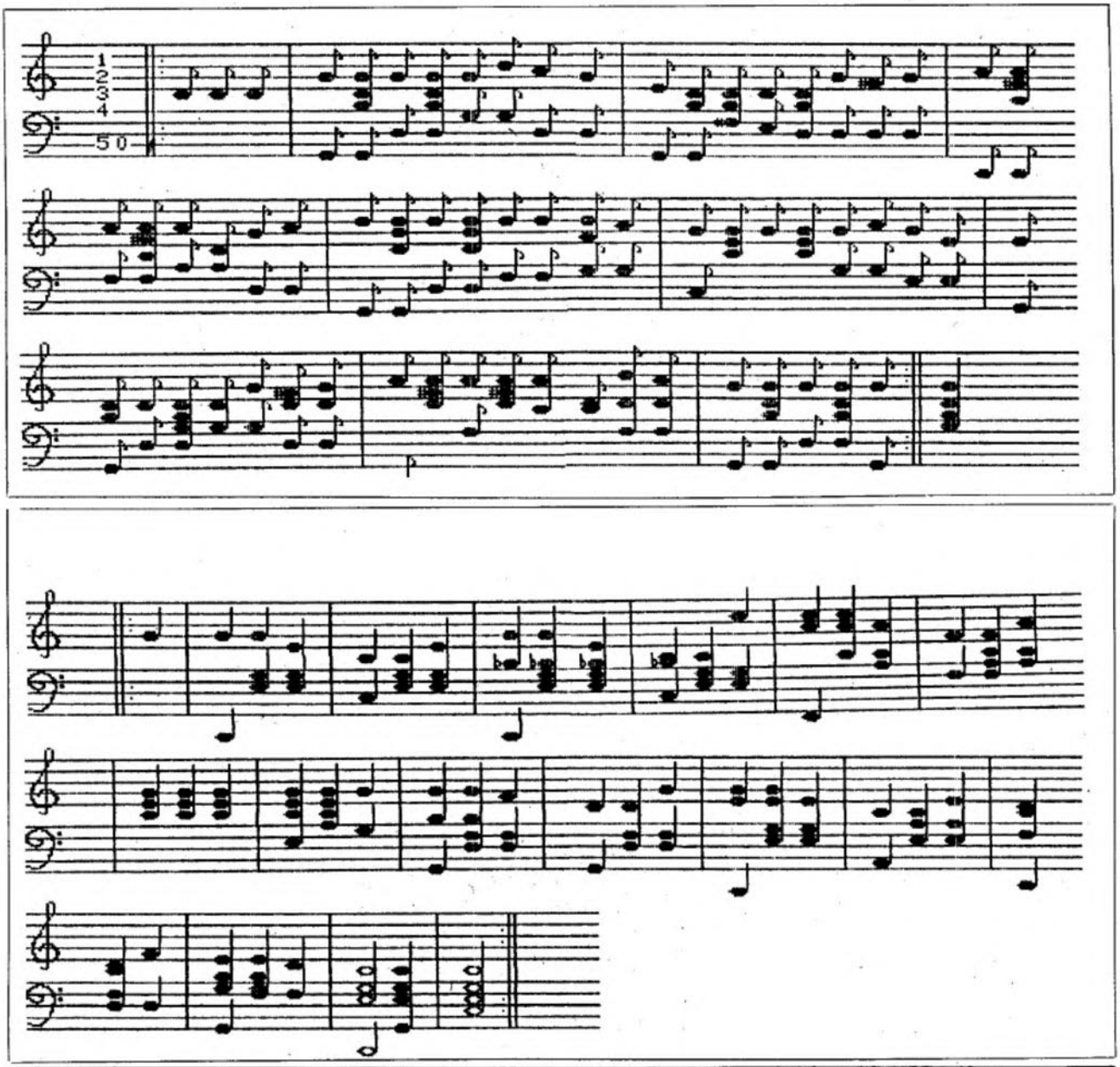
By Mal McLauchlan
MUSIC
32K ECB + MUSIC II

I'M RAPIDY BECOMING fascinated with MUSICA 11 now that I've come to grips with some of its complexities (complex to me anyway as I know little music theory).

It's a program where you see a representation of the actual sheet music on the screen, and have a great degree of control over the tone (trumpet, violin etc.). Different voices in the harmony can be emphasised to provide more flexibility.

Here I have compiled two "folksy" songs, "Last Night I Had The Strangest Dream" and "You Have To Walk That Lonesome Valley".

Incidentally, along with Telewriter 64 and CoCoMax, I have Musica II working well on CoCo 3.



MACDANCE

By Mal McLauchlan
MUSIC
16K ECB

THE SIMPLE ONE-PART music capability of CoCo, without special software, provides a good vehicle for playing certain types of music.

One type that I have targeted on is the flute-like notes of folk music, such as played for Scottish traditional dances (when flutes became bagpipes).

These dances in their original setting were often performed in a Highland glen with a sole piper providing the rhythmic sounds - music?

My greatest musical pleasure comes from attempting to simulate the complex waveforms of various instruments using programs like MUSIC+ and MUSICA II.

But the simple 1-part music isn't far behind when it comes to personal enjoyment.

Anyway play these 5 Scottish dances and see what you think!

By the way, what I'd like to see in Softgold from some "expert" CoCo 3 graphics programmer, is a super-duper tartan screen. I reckon it would look superb.

The Listing:

```
1 '6 TRADITIONAL SCOTTISH DANCES
 ARRANGED BY MALCOLM MCLAUCHLAN
11 HUNTER ST.,BOONAH,Q.4310*****
2 GOTO 10
3 SAVE"295:3":END'7
4 '**REELS, JIGS ETC***********
10 'TRADITIONAL SCOTTISH REEL,
"PUSH ABOUT THE JORUM"**********
20 GOSUB 640
30 PRINT@5," PUSH ABOUT THE JORU
M ";
40 PLAY"P1"
50 FOR RPT=1 TO 2
60 PLAY"T3O4L8GDO3GBGF#AAO4CO3BG
BO4DL16EF#L4GL8F#EDO3GBGF#AAO4CO
3BGAF#L4GL8G"
70 NEXT RPT
80 FOR RPT=1 TO 2
90 PLAY"T4O4L8DL4GL8F#GAF#EDGGF#
GL4AL8DF#L4GL8F#GAF#EDEGF#AL4GL8
GD"
100 PLAY"T4O4L4GL8F#GAF#EDGGF#GL
4AL8DF#L4GL8F#GAF#EDEGF#AG"
110 NEXT RPT
120 PLAY"T5O4L4GL8F#GAF#EDGGF#GL
4AL8DF#L4GL8F#GAF#EDEGF#AGL4.G"
130 '***LORD MACDONALD'S REEL***
140 PRINT@5,"LORD MACDONALD'S RE
EL ";
150 PLAY"P1T4O3L8BO4L4.DL8EDO3BO
4GO3BO4DO3BO4GO3BO4AO3AABO4DC#DE
DO3BO4GO3BL16ABO4L8CO3BABGG"
160 FOR RPT=1 TO 2
170 PLAY"T4O3L8BDGBGAGBGDGBGAGEG
DGBDAGBGDEDCO2BGGO3G"
180 PLAY"T4O3L8DGBGAGBGDGBGAGEGB
AGF#GF#EDEF#GABGG"
190 NEXT RPT
200 PLAY"L4G"
210 '**THE BOB OF FETTERCAIRN**
220 PRINT@5,"THE BOB OF FETTERCA
IRN";
230 PLAY"P1"
240 FOR RPT=1 TO 2
250 PLAY"T3O3L16BO4CL8DO3BGBO4DO
3BGBO4DO3BO4DGF#DF#EC#O3AO4C#EC#
O3AO4C#EDEF#GAGE"
260 PLAY"T3O4L8DO3BGBO4DO3BGBO4D
O3BO4DGL4F#L8DF#AF#GBF#DGO3BL16A
L8.AL8BO4DL4EL8EG"
270 NEXT RPT
280 FOR RPT=1 TO 2
290 PLAY"T3O4L8DGO3BO4GDGO3BO4GD
GO3BO4GL4F#L8DF#EAC#AEAC#AEDEF#G
AGE"
300 PLAY"T3O4L8DGO3BO4GDGO3BO4GD
GO3BO4GL4F#L8DF#AF#GEF#DGO3BL16A
L8.AL8BO4DL4EG"
310 NEXT RPT
320 PLAY"L4G"
330 '**ONE HUNDRED PIPERS*******
340 PRINT@5," ONE HUNDRED PIPER
S ";
350 PLAY"P1"
360 FOR RPT=1 TO 2
370 PLAY"T4O3L4GL8AL4BL8DDEDL4EL
8GL4GO4L8EL4DO3L8BBAGL4BL8AAGAL4
BL8DDEDL4EL8GL4GO4L8EL4DO3L8BABA
L4.GL4G"
380 NEXT RPT
390 FOR RPT=1 TO 2
400 PLAY"T4O3L8BO4L4DL8DDO3BO4DL
4EL8F#GF#EL4DO3L8BBAGL4BL8AL4AL8
BO4L4DL8DDO3BO4DL4EL8F#GF#EL4DO3
L8BABAL4.GL4G"
410 NEXT RPT
420 '**SMASH THE WINDOWS********
430 PRINT@5," SMASH THE WINDOWS
 ";
440 PLAY"P1"
450 FOR RPT=1 TO 2
460 PLAY"T4O3L8ADEDL4F#L8AO4L4DL
8F#EC#O3AL4GL8BL4F#L8AL4EL8F#GF#
D"
470 PLAY"T4O3L8DEDL4F#L8AO4L4DL8
F#EC#O3ABO4GF#EDC#L4.DL4D"
480 NEXT RPT
490 FOR RPT=1 TO 2
500 PLAY"T4O4L8AL4AL8F#L4DL8F#O3
L4AO4L8AGF#L4GL8EL4C#L8EO3L4AO4L
8GGF#E"
510 PLAY"T4O4L4F#L8DL4GL8EL4AL8F
#L4BL8GF#EDEF#C#L4.DL4D"
520 NEXT RPT
530 '**BIODAG AIN N'ONISH*******
540 PRINT@5," BIODAG AIN N'ONIS
H ";
550 PLAY"P1"
560 FOR RPT=1 TO 2
570 PLAY"T4O4L8DC#O3AO4L16C#DL8E
L4C#O3L8AO4C#O3BGO4DO3GL4BL8GBO4
C#O3AO4L16C#DL8EL4C#O3L8AO4C#O3B
GO4DO3BL4AL8A"
580 NEXT RPT
590 FOR RPT=1 TO 2
600 PLAY"T4O4L8G#AEEC#L4AL8EF#GD
DO3BO4L4GL8DG#AEEC#L4AL8EDC#O3AB
G#L4AL8A"
610 NEXT RPT
620 PRINT@5,"         THE END
 ";
630 GOTO 630
640 '*****SUBROUTINE FOR SCREEN
DRESS-UP**************************
650 CLS3:Y=1:FOR X=1 TO 62:SET(X
,Y,4):NEXT X
660 Y=4:FOR X=1 TO 62:SET(X,Y,2)
:NEXT X
670 Y=7:FOR X=1 TO 62:SET(X,Y,4)
:NEXT X
680 Y=10:FOR X=1 TO 62:SET(X,Y,2
):NEXT X
690 Y=13:FOR X=1 TO 62:SET(X,Y,4
):NEXT X
700 Y=16:FOR X=1 TO 62:SET(X,Y,2
):NEXT X
710 Y=19:FOR X=1 TO 62:SET(X,Y,4
):NEXT X
720 Y=22:FOR X=1 TO 62:SET(X,Y,2
):NEXT X
730 Y=25:FOR X=1 TO 62:SET(X,Y,4
):NEXT X
740 Y=28:FOR X=1 TO 62:SET(X,Y,2
):NEXT X
750 Y=31:FOR X=1 TO 62:SET(X,Y,4
):NEXT X
```

Continued on p56

WEATHER REPORT

By Harry Hoffmann

UTILITY

32K COCO

TODAYS PROGRAM IS the start of a series of ten programs dealing with the weather. The first five programs take care of the temperature and the second five are about the rainfall.

To give as many people as possible the chance to use these programs, I have written the majority of them to suit the CoCo 2.

The second program prints a chart to the DMP-105, the third one displays the chart in the CoCo 3 HSCREEN, the fourth program displays the chart in the PMODE 4 screen, and the last one prints a chart to the TP-10 printer.

This first program is a data input program. It creates a datafile, which will be used in the following four programs.

You can input any amount of data, (daily temperatures) you like.

After loading an existing file into the computer, the options offered are renewing or extending the file - don't worry about putting in too much data (eg, 32 days per month).

The other programs will choose the right amount of data. If you don't have enough memory, you can omit all comments and possibly the second page of instructions.

Please save this program as ["3tem-dat/bas"] - thank you very much. I hope these programs are of much use to you.

The Listing:

```
0 GOTO20
1 SAVE"317:3":END'3
2 ********************************
3 ** 3TEM-DAT/BAS PROGRAM  BY **
4 **       HARRY HOFFMANN       **
5 ** CROWS NEST CARAVAN PARK   **
6 ** CROWS NEST QLD 4355       **
7 ********************************
8 '
9 '
10 ****************************
   ** CHANGE <OK>  TO <HH>  **
   ****************************
20 POKE44014,72:POKE44015,72
30 ' ************************
     **  SPECIAL CLS POKE  **
     **  CLS 0 TO 255      **
     ************************

40 FOR I=43288 TO 43302:POKE I,1
8:NEXT I
50 '******************************
    * THIS PROGRAM WAS WRITTEN *
    * ON A COCO 3 BUT SHOULD   *
    * RUN ON ANY  COCO 2 DECB  *
    ******************************

60 CLEAR5000
70 DIM L(31):DIM H(31)
80 DIM D$(75)
90 E$=CHR$(128)
100 A$(1)=E$+"insert"+E$+"disc"+
E$+"any"+E$+"key"+E$
110 A$(2)=" INSERT DISC ANY KEY
"
120 AR$=" I  AM TERRIBLY  SORRY,
 BUT "
130 D=1
140 CLS213:'*NO IT'S NO MISTAKE
-IT IS CLS 213****
150 PRINT@100," TEMPERATURE DATA
 INPUT ";
160 PRINT@171," FOR DISC ";
170 PRINT@238," BY ";
180 PRINT@296," HARRY HOFFMANN "
;
190 PRINT@362," CROWS NEST ";
200 SCREEN0,1
210 GOSUB240:GOTO280
220 ******************************
    * TITLE SONG BY H.HOFFMANN *
    ******************************

230 ** IDEA FROM CRAIG STEWARD-C
OCO MAGAZINE MAY/1987 PAGE 61 --
THANKS CRAIG ***
240       FORX=1TO3:PLAY"T6O3V30L
4CV20L8CV10CL8V30CL16V20CV10CL8V
30DL16V20DV10L8V30EL16V20EV10EL4
V30CL8V16CV10CL4V30FL8V20FV10FL4
V30EL8V20EV10EL4.V30DL8.V20DV10D
":NEXT
250       FORX=1TO3:PLAY"V30L8GV2
0L16GV10G":NEXT:PLAY"V30L4.GL8.V
20GV10GV30L8FV20L16FV10FV30L4EV2
0L8EV10EL4V30DL8V20DV10DL2V30GL4
V20GV10G"
260       PLAY"V30L8GV20L16GV10GV
30L4.GV20L8.GV10GV30L8FV20L16FV1
0FV30L4EV20L8EV10EL4V30DL8V20DV1
0DL1V30CL4V20CV15CL2V10CV5C"
270 RETURN
280 CLS249:GOSUB1850
290 PRINT@224,"DO YOU NEED INSTR
UCTIONS?  Y/N  ";
300 I$=INKEY$
310 IFI$=""THENPRINT@251," / ":F
ORX=0TO300:NEXT:PRINT@251,"Y/N";
:FORX=0TO300:NEXT
320 IFI$="Y"THEN1870
330 IFI$<>"N"THEN300
340 CLS230
350 GOSUB1850
360 PRINT@98,"INPUT MONTH AND YE
AR (MM/YY)";
370 PRINT@162,"";:INPUT DT$:PRIN
T@190,STRING$(2,PEEK(1533));
380 DT=VAL(LEFT$(DT$,2))
390 IF DT<1 OR DT>12 THEN PRINT@
418,AR$;:PRINT@450," A YEAR HAS
ONLY 12 MONTHS. ";:PRINT@482," P
LEASE GIVE IT ANOTHER GO. ";:GOS
UB1500:GOTO340
400 IF LEN(DT$)<>5THEN PRINT@386
,AR$;:PRINT@418," THE DATE YOU'V
E JUST GIVEN ";:PRINT@450," ME H
AS NOT THE RIGHT LENGTH";:PRINT@
482," PLEASE TRY    ONCE  MORE.
 ";:GOSUB1500:GOTO340
410 PRINT@258," IS THE DATE CORR
ECT - "DT$;
420 PRINT@354," Y/N ";
430 I$=INKEY$:IF I$=""THEN PLAY"
T255V15C":GOTO430
440 IF I$="N"THEN340
450 IF I$<>"Y"THEN SOUND100,5:GO
TO 340
460 CLS234
470 GOSUB1850
480 PRINT@229,A$(1);:FORZ=1TO300
:NEXT
490 PRINT@229,A$(2);:FORZ=1TO295
:NEXT
500 IF INKEY$=""THEN480
510 F$=LEFT$(DT$,2)+"*"+RIGHT$(D
T$,2)+"TEM"
520 '** CHECK DISC FOR FREE GRAN
ULES **
530 IF FREE(0)<2 THEN CLS181:GOS
UB1850:PRINT@161,"SORRY,NOT ENOU
GH SPACE ON DISC";:GOTO540 ELSE
GOTO 570
540 PRINT@227," PLEASE INSERT OT
HER DISC ";
550 PRINT@289,"AND PRESS ANY KEY
 TO CONTINUE ";
560 EXEC44539:GOTO530
570 GOSUB1520:IF D<>1THEN1770
580 CLS145
```

```
590 GOSUB1850
600 '** FORCED FILE INPUT ROUTIN
E TO EXAMINE FILE BEFORE MESSING
 IT UP **
610 OPEN "I",#1,F$
620 PRINT@98," FILE ";F$;" open
";
630 IF EOF(1)=-1THEN720
640 FOR S=1TO31
650 IF EOF(1)=-1THEN720
660 INPUT#1, L(S)
670 INPUT#1, H(S)
680 PRINT@226," TEMP.FOR ";:PRIN
TUSING"##";S;:PRINT"/";DT$;" = "
;
690 PRINT@290," LOW  = ";:PRINTU
SING"###";L(S);:PRINT@354," HIGH
 = ";:PRINTUSING"###";H(S);
700 FOR DL=1TO700:NEXTDL
710 NEXTS
720 CLOSE#1
730 PRINT@98," FILE ";F$;" close
d ";
740 IF S<1THEN S=1
750 PRINT@162,S-1;" ENTRIES ";
760 FORX=1408TO1535:POKEX,241:NE
XTX
770 PRINT@418," [E] = EXTEND FIL
E  ";
780 '
790 '** ATTENTION **
800 '** THIS INPUT <N> WILL ERRA
SE EXISTING FILE AND START THE S
AME AT RECORD NO.1 **
810 PRINT@450," [N] = NEW FILE
   ";
820 PRINT@482," [Q] = QUIT PROGR
AM ";
830 I$=INKEY$:IF I$=""THEN PLAY"
O5T255V31B":GOTO830
840 IF I$="E"THEN PLAY"O3":GOTO9
50
850 IF I$<>"N"THEN920
860 CLS181:GOSUB1850:PRINT@106,"
 attention ";:PRINT@162," TO CON
TINUE FROM HERE WITH ";:PRINT@19
4," A <Y> WILL UTTERLY DESTROY "
;:PRINT@226," THE ALREADY EXISTI
NG FILE. ";
870 PRINT@258," IF YOU WISH TO D
O SO,PRESS ";:PRINT@290," <Y> OT
HERWISE PRESS <N> ** ";
880 I$=INKEY$:IFI$=""THENPLAY"T2
55V31O4C":GOTO880
890 IFI$="Y"THENT=1:GOTO1300
900 IFI$<>"N"THEN880
910 CLS233:GOSUB1850:GOTO770
920 IF I$<>"Q"THEN830
930 GOTO1810
940 '**FILE EXTENSION ROUTINE**
950 CLS198:GOSUB1850
960 OPEN"I",#1,F$
970 PRINT@98," FILE ";F$;" open
<I>";
980 OPEN"O",#2,"TEMPTEMP/DAT"
990 PRINT@130," FILE TEMPTEMP op
en <O>";
1000 FOR T=1TO S-1
1010 IF LOF(1)=-1THEN1070
1020 INPUT#1, L(T)
1030 PRINT#2,L(T)
1040 INPUT#1, H(T)
1050 PRINT#2,H(T)
1060 NEXTT
1070 CLOSE#1
1080 PRINT@98," FILE ";F$;" clos
ed<I>";
1090 PRINT@194,E$"input"E$"new"E
$"data"E$"now"E$"please"E$;
1100 PRINT@258," low TEM.FOR ";:
PRINTUSING"##";T;:PRINT"/";:PRIN
TDT$;:PRINT" = ";:INPUT L(T):PRI
NT@286,STRING$(2,PEEK(1024));
1110 PRINT@322," IS THIS CORRECT
? -";L(T);"Y/N";
1120 I$=INKEY$:IF I$=""THEN1120
1130 IF I$="N"THENPRINT@256,STRI
NG$(96,198);:GOTO1100
1140 IF I$<>"Y"THEN1120
1150 PRINT@386,"high TEM.FOR ";:
PRINTUSING"##";T;:PRINT"/";:PRIN
TDT$;:PRINT" = ";:INPUT H(T):PRI
NT@414,STRING$(2,PEEK(1024));
1160 PRINT@450," IS THIS CORRECT
? -";H(T);"Y/N";
1170 I$=INKEY$:IF I$=""THEN1170
1180 IF I$="N"THENPRINT@384,STRI
NG$(96,198);:GOTO1150
1190 IF I$<>"Y"THEN1170
1200 PRINT#2, L(T)
1210 PRINT#2, H(T)
1220 PRINT@257,STRING$(250,198);
1230 PRINT@357," ANY MORE DATA ?
  Y/N ";:SCREEN0,1
1240 I$=INKEY$:IF I$=""THEN1240
1250 IF I$="N"THEN1700
1260 IF I$<>"Y"THEN1240
1270 PRINT@357,STRING$(25,198);
1280 T=T+1:GOTO1100
1290 '** NEW FILE ROUTINE **
1300 CLS150:GOSUB1850
1310 T=1:OPEN"O",#1,F$
1320 PRINT@98," NEW FILE ";F$;"
open <O> ";
1330 PRINT@162,E$"input"E$"data"
E$"now"E$"plaese"E$;
1340 PRINT@226," low TEM.FOR ";:
PRINTUSING"##";T;:PRINT"/";DT$"
= ";:INPUT L(T):PRINT@254,STRING
$(2,PEEK(1535));
1350 PRINT@290,"high TEM.FOR ";:
PRINTUSING"##";T;:PRINT"/";DT$"
= ";:INPUT H(T):PRINT@318,STRING
$(2,PEEK(1535));
1360 PRINT@354," IS THIS CORRECT
 ?  Y/N ";
1370 I$=INKEY$:IF I$=""THEN1370
1380 IF I$="N"THEN PRINT@226,STR
ING$(200,PEEK(1024));:GOTO1340
1390 IF I$<>"Y"THEN1370
1400 PRINT#1,L(T)
1410 PRINT#1,H(T)
1420 PRINT@418," ANY MORE DATA
Y/N ";
1430 I$=INKEY$:IF I$=""THEN1430
1440 IF I$="Y"THEN T=T+1:PRINT@2
26,STRING$(250,PEEK(1024));:GOTO
1340
1450 IF I$<>"N"THENSOUND100,10:G
OTO1430
1460 CLOSE#1:CLS138
1470 PRINT@103," ALL FILES CLOSE
D ";:GOSUB1500
1480 GOTO1810
1490 ** DELAY ROUTINE **
1500 FORX=1TO4000:NEXT:RETURN
1510 ** CHECK-DISC-FOR-FILE ROUT
INE** THANKS TO BILL TOTTINGHAM
FOR THIS ONE ** RAINBOW APRIL/19
86, PAGE 18-20 **
1520 FOR X=3TO11
1530 DSKI$ 0,17,X,A$,B$
1540 C$=A$:GOSUB1620
1550 IF D$(F)=F$ THEN D=1:RETURN
1560 C$=B$:GOSUB1620
1570 IF D$(F)=F$ THEN D=1:RETURN
1580 IF LEFT$(A$,8)=F$ THEN RETU
RN
1590 NEXTX
1600 CLS154:GOSUB1850:PRINT@226,
" FILE ";F$;" DOESN'T EXIST";
1610 D=0:RETURN
1620 FORM=1TO128STEP32
1630 F=F+1
1640 D$(F)=MID$(C$,M,8)
1650 IF D$(F)=F$ THEN RETURN
1660 NEXTM
1670 RETURN
1680 '**FINISHING ROUTINE FOR**
1690 '**EXTENDING FILE        **
1700 CLS214:GOSUB1850
1710 CLOSE#2
1720 PRINT@194,"       ALL FILES
CLOSED       ";
1730 PRINT@258," RENAME FILE NOW
 PLEASE WAIT";
1740 G$=F$+"/DAT"
1750 KILL G$:RENAME"TEMPTEMP/DAT
"TO G$
1760 GOSUB1500:GOTO1810
1770 PRINT@418," WONNA CREATE NE
W FILE  Y/N ";
1780 I$=INKEY$:IF I$=""THEN1780
1790 IF I$="Y"THEN T=1:GOTO1300
1800 IF I$<>"N"THEN1780
1810 CLS138:GOSUB1850
1820 PRINT@235,E$"finish"E$;:PRI
NT@416,"";
1830 GOSUB240:END
1840 ***  TITLE ROUTINE ***
1850 PRINT@32,STRING$(2,128)"tem
perature"CHR$(128)"by"CHR$(128)"
harry"CHR$(128)"hoffmann"CHR$(12
8);
1860 RETURN
1870 CLS154:GOSUB1850
1880 PRINT@106,"instructions";
1890 PRINT@160,"THIS PROGRAM CRE
ATES A PERMANENTFILE ON DAILY HI
GH-, AND LOW-   TEMPERATURES. IT
 IS FAIRLY FOOL PROOF. YOU CAN P
UT IN ANY AMOUNTOF DATA YOU LIKE
. EG.DAILY ---  WEEKLY --- EVERY
 10 DAYS OR ONCEA MONTH. JUST FO
LLOW THE SIMPLE ";
1900 PRINT@384,"INSTRUCTIONS CAR
EFULLY AS YOU GOALONG.
                PRESS ANY KEY TO
 CONTINUE       ";:EXEC44539
1910 PRINT@160,"THIS IS THE FIRS
T OF A SERIES OFFIVE PROGRAMS DE
```

Continued on p14

GRAPHIC TO DATA

By Tom Lehane

UTILITY
32K ECB

A GREAT MANY graphic screen pictures are composed using special drawing programs written in either Basic or machine language. The art work from these drawing programs are saved in binary format and can only be viewed by loading back into the computer using the load function from the drawing program or by loading and writing a small program to view them.

The small Basic program consists of two lines as follows:

```
10 PMODE4:SCREEN1,1
20 GOTO 20
```

This will display a M/L screen loaded back into the computer. You can use this small program to view a graphic screen of your favourite game.

Load your M/L game and type EXEC. When a graphic screen is displayed, press the RESET button and write the two line program above and type RUN.

Your game screen should now be seen and held for viewing by the GOTO in line 20.

You can save this game screen using the "SCR SAVE" program accompanying this article.

Please note that the above instruction will only work if the machine language program or game does not auto-start.

The idea behind auto-start software was to make the program easy to load and use.

Software programmers soon found it also stopped piracy of their product and was included in a lot of software made available for the CoCo.

Some copy programs could get around this protection, but in general it was a deterrent.

To have the art work printed in a magazine or sent to a bulletin board over the modem is not possible unless it is in ASCII format.

The program, "GRAFDATA", converts a PMODE graphic screen to a Basic program in the form of HEX$ data statements.

The compiler creates a ready-to-run basic program that is saved to either Disk or Tape.

The GRAFDATA program provides compatibility with tape or disk users, so there is no need to modify any part of the compiling program.

The program PEEKs the start of the current HI-RES screen so it does not matter if you are using a tape or disk system.

This PEEK is also in the compiled Basic program so it can be loaded and RUN even if used by a tape based user and the program was compiled from a disk system.

To get GRAFDATA to compile a graphic screen, you can either load a Basic program that has your chosen graphic picture and RUN the program so it will draw the picture to one of the PMODE screens. Then press the reset button and load GRAFDATA.

This will over-write the previous Basic program but will leave the graphic screen untouched.

Follow the on-screen prompts to view, change PMODE or compile the graphic screen.

The program provides for loading a M/L graphic screen from both tape or disk.

The only thing to note is that a M/L screen saved from disk will not load in to the correct PMODE location from a tape based set up as disk basic uses the normal PMODE area for it's own operations and both graphic PMODE areas start are at different addresses. Tape based user's HI-RES screens starts from 1536 and disk starts from 3584 so to load a M/L saved screen from disk in to a tape based system the program would be loaded 2048 bytes above the normal start address.

For the same token if a tape saved screen was loaded in to a disk system the M/L screen would be loaded onto disk Basic's operations area.

Once GARFDATA has compiled your graphic screen into a Basic data program, it can then be loaded into both tape and disk systems and RUN without any further changes.

Due to the large amount of memory to be compiled from a PMODE 4 graphic screen, the program can be a little slow. If detail is not of great importance the graphic screen could be compiled from PMODE 0 to save time when sending data over the modem.

NOTE: The screen would have to be drawn in PMODE 0 not compressed from PMODE 4.

SCR SAVE program ....

```
10 REM HI-RES SAVE BY TOM LEHANE
20 REM ----------------------
30 REM PEEK FOR GRAPHIC AREA
40 REM ----------------------
50 ST=PEEK(186)*256+PEEK(187)
60 EN=PEEK(183)*256+PEEK(184)
70 REM --------------
80 IF EN-ST=1536THEN M=1ELSEM=4
90 REM --------------
100 REM * SHOW PIX *
110 REM --------------
120 PMODEM:SCREEN1,1
130 FOR X=1 TO 1800:NEXTX
140 CLS:PRINT@160,"DO YOU WANT T
O CHANGE MODE?"
150 PRINT"IF SO E NTER PMODE (0
TO 4)"
160 PRINT"ELSE PRESS ENTER"
170 A$=INKEY$:IF A$=""THEN 170
180 IF A$=CHR$(13) THEN 220
190 M=VAL(A$)
200 IF M>4 THEN140
210 GOTO 120
220 EN=PEEK(183)*256+PEEK(184)
230 PRINTSTRING$(32,131);
240 PRINT"DO YOU WISH TO SAVE SC
REEN TO"
250 INPUT" (T)APE OR (D)ISK";DV$
260 PRINTSTRING$(32,131);
270 INPUT" FILE NAME";F$
280 IF DV$="D" THEN 300
290 CSAVEMF$,ST,EN,ST:END
300 F$=F$+"/BIN"
310 SAVEMF$,ST,EN,ST:END
```

The Listing:

```
0 GOTO10
3 SAVE"325:3":END'8
10 REM ----------------------
20 REM GRAFDATA BY TOM LEHANE
30 REM AUGUST 1987
40 REM ----------------------
50 CLEAR 1000
60 GOTO 190
70 REM ------------------
80 REM * SUB ROUTINE AREA *
90 REM ------------------
100 FOR X=A TO B
110 POKEX,PEEK(X)-64
120 NEXT X:RETURN
130 PRINT#DV,A$:PRINTA$:RETURN
140 PRINTSTRING$(32,131);:RETURN
150 A$=INKEY$:IF A$=""THEN150 EL
SE RETURN
160 REM ------------------
170 REM * TITLE SCREEN *
180 REM ------------------
190 CLS:PRINT TAB(11)"GRAFDATA"
200 PRINT TAB(8)STRING$(15,39)
210 PRINT"CONVERTS A BINARY GRAP
HIC SCREEN TO DATA."
220 PRINT TAB(3)"NEED INSTRUCTIO
NS ( Y= YES)"
230 PRINT@228,CHR$(139)+CHR$(129
)
240 PRINT@262,CHR$(132)
250 PRINT@264,"PROGRAM BY TOM LE
HANE"
260 A=1024:B=1215
270 GOSUB 100
280 GOSUB 150
290 IF A$="Y" THEN 1230
300 GOSUB 140
310 PRINT@224,STRING$(32,141)
320 PRINT TAB(4)"DO YOU WISH TO
LOAD FROM"
330 PRINT TAB(8)"(T)APE OR (D)IS
K"
340 PRINT TAB(8)"ELSE PRESS ENTE
R"
350 INPUT"--->";DV$
360 IF DV$="T" THEN 1140
370 IF DV$="D" THEN 1170
380 REM ----------------------
390 REM PEEK FOR GRAPHIC AREA
400 REM ----------------------
410 ST=PEEK(186)*256+PEEK(187)
420 EN=PEEK(183)*256+PEEK(184)
430 REM SET MODE TO VALUE OF M
440 IF EN-ST=1536THEN M=1ELSEM=4
450 REM --------------
460 REM * SHOW PIX *
470 REM --------------
480 PMODEM:SCREEN1,1
490 FOR X=1 TO 1800:NEXTX
500 CLS:PRINT@160,"DO YOU WANT T
O CHANGE MODE?
510 PRINT"IF SO ENTER PMODE (0 T
O 4)"
520 PRINT"ELSE PRESS ENTER"
530 A=1184:B=1279:GOSUB 100
540 GOSUB 150
550 IF A$=CHR$(13) THEN 590
560 M=VAL(A$)
570 IF M>4 THEN 500
580 GOTO 480
590 EN=PEEK(183)*256+PEEK(184)
600 GOSUB 140
610 PRINT"DO YOU WISH TO SAVE DA
TA TO"
620 INPUT" (T)APE OR (D)ISK";DV$
630 IF DV$="T" THEN DV=-1
640 IF DV$="D" THEN DV=1
650 GOSUB 140
660 INPUT"FILE NAME";N$
670 N$=LEFT$(N$,8)
680 IF DV=1 THEN N$=N$+"/BAS"
690 GOSUB 140
700 PRINT"PRESS ANY KEY WHEN I/O
 DEVICE IS READY"
710 GOSUB 150
720 REM ----------------------
730 REM * OPEN FILE AND COMPILE
740 REM * GRAPHIC PIX TO DATA
750 REM ----------------------
760 OPEN"O",#DV,N$
770 HX$=CHR$(34)+"&H"+CHR$(34)
780 A$="1 PMODE"+STR$(M)+",1:PCL
S:SCREEN1,1":GOSUB 130
790 A$="2 N=PEEK(186)*256+PEEK(1
87):L=PEEK(183)*256+PEEK(184)
800 GOSUB 130
810 A$="3 FOR T=N TO L:READX$"
820 GOSUB 130
830 A$="4 X=VAL("+CHR$(34)+"&H"+
CHR$(34)+"+X$):POKE T,X:NEXT T"
840 GOSUB 130
850 A$="5 GOTO 5"
860 GOSUB 130
870 T=10
880 FOR X=ST TO EN STEP75
890 Y=X:IF (EN-X)<75 THEN N=EN-X
+1 ELSE N=75
900 FOR P=1TO N
910 IF P=N THEN Z$=Z$+HEX$(PEEK(
Y)):CLS:GOSUB 1110:GOTO930
920 Z$=Z$+HEX$(PEEK(Y))+","
930 Y=Y+1
940 NEXTP
950 T$=STR$(T)+" DATA "
960 A$=T$+Z$
970 GOSUB 130
980 T=T+1
990 Z$=""
1000 IF X=EN THEN 1010 ELSE NEXT
 X
1010 CLOSE #DV
1020 REM ----------------------
1030 REM RESTART GRAFDATA OR END
1040 REM ----------------------
1050 CLS:GOSUB1110:PRINT TAB(9)"
NOW COMPLETED":GOSUB 140
1060 PRINT"RESTART GRAPHIC TO DA
TA PROGRAM"
1070 PRINT TAB(10)" Y = YES"
1080 GOSUB 150
1090 IF A$="Y" THEN 300 ELSE END
1100 GOSUB 150
1110 PRINT TAB(3)"CONVERTING GRA
PHIC PICTURE"
1120 PRINT TAB(3)STRING$(26,131)
1130 RETURN
1140 PRINT@160,"NAME OF PROGRAM
TO LOAD":GOSUB 140
1150 INPUT"--->";N$:CLOADMN$
1160 GOTO 410
1170 CLS:DIR
1180 PRINT:PRINT"INPUT NAME OF P
ROGRAM TO LOAD"
1190 PRINT"INCLUDE EXTENTION":IN
PUTN$
1200 LOADM N$
1210 POKE&HFF40,0:GOTO410
1220 '
1230 CLS
1240 PRINT"          GRAPHIC TO DA
TA":PRINT@32,STRING$(32,131);
1250 PRINT"GRAFDATA CONVERTS A P
MODE SCREEN";:PRINT"TO A BASIC P
ROGRAM IN THE FORM"
1260 PRINT"OF HEX$ DATA STATEMEN
TS. THE":PRINT"COMPILER CREATES
A READY TO RUN"
1270 PRINT"BASIC PROGRAM THAT IS
 SAVED TO":PRINT"TAPE OR DISK. T
HE 'GRAFDATA'"
1280 PRINT"PROGRAM PROVIDES COMP
ATIBILITY":PRINT"WITH TAPE OR DI
SK USERS SO THERE";
1290 PRINT"IS NO NEED TO MODIFY
ANY PART OF";:PRINT"THE COMPILED
 BASIC PROGRAM. THIS";
1300 PRINT"MAKES GRAFDATA USEFUL
 FOR MODEM":PRINT"USE."
1310 PRINT"     how to use .....
....."
1320 PRINT@484,"PRESS ENTER TO C
ONTINUE";:LINEINPUTZZ$
1330 CLS
1340 PRINT"          GRAPHIC TO DA
TA":PRINT@32,STRING$(32,131);
1350 PRINT"FIRST YOU NEED A GRAP
HIC PICTURE";:PRINT"ON ONE OF TH
E PMODE SCREENS."
1360 PRINT"THIS CAN BE RUN FROM
A BASIC":PRINT"PROGRAM THAT DRAW
S A PICTURE OR"
1370 PRINT"A SCREEN SAVED IN BIN
ARY (M/L).":PRINT"GRAFDATA ALLOW
S YOU TO CHANGE"
1380 PRINT"THE PMODE TO SUIT THE
 GRAPHIC":PRINT"PICTURE JUST LOA
DED AS BASIC"
1390 PRINT"CAN NOT DETERMINE WHI
CH PMODE":PRINT"TO ADDRESS, FROM
 A M/L LOAD."
1400 PRINT"FOLLOW ON SCREEN PROM
PTS TO":PRINT"COMPILE GRAPHIC SC
REEN."
1410 PRINT
1420 PRINT@484,"PRESS ENTER TO C
ONTINUE";:LINEINPUTZZ$
1430 CLS:GOTO 300
```

■

TAPE READER

By Malcolm Patrick

UTILITY
32K ECB COCO

I AM A HORDER of programs. Like most CoCo nuts, I save everything on tape because that is the safest place to store programs for any length of time.

Over the months I have accumilated many tapes and consequentially keep forgetting what was on the tapes.

Well that problem is now solved. I have written a program that will read the names and types of the the program and then save these titles to disk.

These titles are saved in an ASCII format and are then read in with an ASCII-read program. I use Telewriter-64 for my ASCII-read program.

MAKE SURE the LAST title on your tape is named "END/BAS", as this will allow you to go into the save mode.

The program is self-prompting and simple to use. I welcome any suggestions that you might like to make, through the magazine.

The Listing:

```
0 GOTO10
1 '******** TAPE READ **********
   **** MALCOLM PATRICK ********
3 SAVE"326:3":END'8
10 CLS(4):PRINT@41,"* TAPE READ
*";
20 PRINT@77,"* BY *";
30 PRINT@102,"* MALCOLM PATRICK
*";
40 PRINT@136,"* 8 NEWTON ST *";
50 PRINT@170,"* WHYALLA *";
60 PRINT@200,"* P 086457637 *";
70 PRINT@460,"< enter >";
80 I$=INKEY$:IF I$=""THEN 80
100 CLEAR2000
110 DIM B$(500),AA$(500),C$(500)
,D(500),C(500)
115 CLS:PRINT@160,"THIS PROGRAM
WILL READ THE TITLE FROM TAPE AN
D SAVE IT TO DISK     IT CAN THEN
 BE READ IN WITH","     A DATA RE
ADING PROGRAM","       I USE TELE
WRITER64."
120 GOSUB580:PRINT"MAKE SURE THI
S TAPE <<<ENDS>>>   WITH THE FIL
E ** END/BAS **":GOSUB580:PRINT"
WHAT IS THE NAME OF THIS TAPE":I
NPUT A$:GOSUB 400:GOSUB420
130 K=-2
140 IF K=0 THEN G=0 ELSE G=15
150 POKE126,1:POKE 127,218'set u
p read buffer
160 EXEC 42753 'read in block ($
A701)
170 A=PEEK(124):IF A=255 THEN 37
0'  check eof
180 IF A<>0 AND N=0 THEN C$(C)=C
$(C)+" error "
190 IF A<>0 THEN N=N+1:GOTO160
200 IF N<>0 THEN A$=""
210 N=1
220 REM print out filename
230 FOR I=474 TO 481
240 A$=CHR$(PEEK(I)):GOSUB400
250 NEXT I
260 FT=PEEK(482)
270 IF FT=0 THEN A$="    BASIC"
280 IF FT=1 THEN A$="    DATA "
290 IF FT=2 THEN A$="    BINAR"
300 IF FT=3 THEN A$="    ASSEM"
310 IF FT=4 THEN A$="    EDIT "
320 IFFT>4 THENA$="    UNKNOWN"
330 GOSUB400:GOSUB420
340 IF H<>255 THEN 390
350 GOSUB400:GOSUB420
360 GOTO160
370 GOTO160
380 REM skip for cont. file
390 EXEC 42705:N=-1:GOTO370
400 B$=B$+A$:RETURN
410 RETURN
420 C=C+1
430 C$(C)=B$
440 FOR D=1 TO C
450 IF LEFT$(B$,3)="END"THEN 490
460 PRINTC$(D):NEXT D
470 B$="":RETURN
480 RETURN
490 GOSUB580:PRINT"WHAT IS THE N
AME OF DISK FILE   USE ONLY 8 LE
TTERS":INPUT W$
500 GOSUB580:PRINT" PLACE MASTER
 DISK IN DRIVE"," AND < ENTER >"
510 I$=INKEY$:IF I$=""THEN 510
520 IF I$="2"THEN 530 ELSE 530
530 OPEN"O",#1,W$
540 FOR D=1 TO C
550 WRITE #1,C$(D)
560 NEXT D:A$="":C$="":B$=""
570 CLOSE #1:RUN
580 PRINT:PRINT"****************
**************":PRINT:RETURN
```

■

From p11

```
ALING WITH THE  TEMPERATURE. TWO
 DISPLAY (ONE   FOR COCO 2 AND O
NE FOR COCO 3)  AND TWO PRINTER
(DMP105 AND     TP-10) PROGRAMS
ARE FOLLOWING.  ALL ARE USING TH
E FILES CREATED ";
1920 PRINT@384,"BY THIS PROGRAM.
 SO BE PATIENT  AND START COLLEC
TING DATA.      ";:SCREEN0,1:EXE
C44539
1930 GOTO340
1940
################################
## WRITTEN BY     (JULY  1987) ##
## HARRY HOFFMANN              ##
## CROWS NEST CARAVAN PARK     ##
## CROWS NEST QLD 4355         ##
################################

1950
********************************
** ALL COMMENT LINES WITH OR  **
** WITHOUT <'> CAN BE OMITTED **
********************************
```

HINT:

A "SKIPF" for disk

Want to verify the existance of a program on disk without having to type 'DIR', 'LOAD' or whatever from a Basic program? Then try this below, typing it in EXACTLY as shown below.

NOTE: This will work only on a DECB 1.0/2.0 DOS. IF you have a DECB 1.1/2.1 DOS, then change '51338' to '51512'.

Also change '&HC65F' to '&HC68C'.

```
A$="filename/ext":EXEC 51338 A$:
EXEC &HC65F:A=PEEK(&H973)
```

DISKLOCK

By Harry Smith

UTILITY
COCO 3 + DISK

DISKLOCK IS A utility I wrote to prevent unauthorized access at any information stored on a disk.

This program has many benefits and can be used in many ways. For example,

* It could prevent unauthorized access to programs being developed.

* A parent could lock up all the games until the children have completed their homework.

* It could be used to protect home financial and budget files, and ...

* Teachers could use the program to secure test files and test results.

Just about everyone has something they would like to secure!

When the program is first run you are presented with the title page - just follow the prompts.

Instructions and a program description are included within the program. Two passwords are required to operate the progam and are requested before the main menu can be used.

As a Basic program this causes a problem as anybody can list your program to obtain the passwords and unlock protected files. The best method I have found to make the program secure is to place your passwords in line 380 and use a program called "Loadmask" by Glen Dahlgren which is contained on page 24 of the February 1987 edition of American Rainbow Magazine.

This program is a machine-language loader that fools the computer into thinking that the Basic program in memory is machine language. The program also encrypts the Basic program which stops the Basic programs being listed or LLISTed to a printer after it is loaded.

The program contains POKEs to disable the LIST command and POKE113,0 which will perform a cold start if the reset button is pressed. The combination of these POKEs, the ON BRK GOTO command and the encryption by "Loadmask" make this a very secure program (althogh it could probably be broken into by a very good programmer).

I hope that you can find some way in which to use this program - I have used it myself on many disks without any program loss or disk crashes.

I would, however suggest that you should only use the program on backup disks, not your originals.

No liability will be accepted for any program loss as a result of use of this program.

I also welcome improvements and enhancements to the program. If you do make any changes, please don't keep them for yourself, send them in to Softgold magazine so that we all may take advantage of your ideas.

The Listing:

```
0 '********** DISKLOCK *********
 ******* HARRY SMITH ***********
1 GOTO10
3 SAVE"330:3":END'9
10 CLEAR2000
20 ON BRK GOTO 810
30 POKE383,62:POKE113,0
40 PALETTE3,0:POKE65497,0
50 WIDTH40:HSCREEN2:HCLS3:PALETT
E1,39:HCOLOR1:HLINE(50,10)-(280,
30),PSET,BF:HCOLOR14:HPRINT(10,2
),"  The Disk Protector.":HCOLOR2
:HLINE(52,13)-(278,27),PSET,B
60 HCOLOR1:HLINE(50,42)-(280,62)
,PSET,BF:HCOLOR14:HPRINT(9,6),"
   By Harry Smith.":HCOLOR2:HLIN
E(52,45)-(278,59),PSET,B
70 HCOLOR1:HLINE(50,73)-(280,93)
,PSET,BF:HCOLOR14:HPRINT(9,10),"
 Copyright (c) 1987.":HCOLOR2:
HLINE(52,76)-(278,90),PSET,B
80 HCOLOR1:HLINE(50,145)-(280,16
5),PSET,BF:HCOLOR14:HPRINT(8,19)
,"Press <Enter> to Continue.":HC
OLOR2:HLINE(52,148)-(278,162),PS
ET,B90 A$=INKEY$:IFA$="" THEN 90
100 IF A$<>CHR$(13) THEN 100 ELS
E 110
110 HCLS3:PALETTE1,54:HCOLOR1:HL
INE(50,50)-(280,70),PSET,BF:HCOL
OR2:HLINE(52,53)-(278,67),PSET,B
:HCOLOR1:HLINE(50,80)-(280,100),
PSET,BF:HCOLOR2:HLINE(52,83)-(27
8,97),PSET,B:HCOLOR14:HPRINT(7,7
),"Do You Require Instructions."
120 HPRINT(12,11),"Enter < Y or
N >"
130 A$=INKEY$:IF A$="" THEN 130
140 IF A$="Y" OR A$="y" THEN 150
 ELSE IF A$="N" OR A$="n" THEN 3
20
150 HCLS3:PALETTE1,17:HCOLOR1:HL
INE(50,10)-(280,30),PSET,BF:HCOL
OR2:HLINE(52,13)-(278,27),PSET,B
160 HCOLOR14:HPRINT(14,2),"Instr
uctions."
170 HCOLOR1:HLINE(50,40)-(280,15
0),PSET,BF
180 HCOLOR2:HLINE(52,43)-(278,14
7),PSET,B
190 HCOLOR14:HPRINT(7,6),"This P
rogram will allow you":HPRINT(7,
7),"to protect your Disks from":
HPRINT(7,8),"Unautherized access
.":HPRINT(7,10),"If you choose t
o proctet a"
200 HCOLOR14:HPRINT(7,11),"Disk,
 the Directory will be":HPRINT(7
,12),"copied to Track 34 and the
"
210 HCOLOR14:HPRINT(7,13),"origi
nal Directory will be":HPRINT(7,
14),"CRASHED and will show the":
HPRINT(7,15),"word *LOCKED* when
 the DIR ":HPRINT(7,16),"command
 is executed from":HPRINT(7,17),
"Basic."
220 HCOLOR1:HLINE(50,160)-(280,1
80),PSET,BF
230 HCOLOR2:HLINE(52,163)-(278,1
77),PSET,B
240 HCOLOR14:HPRINT(12,21),"Cont
inue - < Y/N >"
250 A$=INKEY$:IF A$="" THEN 250
260 IF A$="Y" OR A$="y" THEN 270
 ELSE  IF A$="N" OR A$="n" THEN
810
270 HCOLOR1:HLINE(50,40)-(280,15
0),PSET,BF:HCOLOR2:HLINE(52,43)-
(278,147),PSET,B
280 HCOLOR14:HPRINT(7,7),"To Unl
ock a *LOCKED* Disk ":HPRINT(7,8
),"Run this program again and":H
```

```
PRINT(7,9),"and Select the Resto
re":HPRINT(7,10),"Option from th
e Menu."
290 HCOLOR14:HPRINT(7,12),"Two P
asswords are required":HPRINT(7,
13),"to continue and will be":HP
RINT(7,14),"requested for entry
on":HPRINT(7,15),"the next scree
n."
300 A$=INKEY$:IFA$=""THEN300
310 IF A$="Y" OR A$="y" THEN 320
 ELSE IF A$="N" OR A$="n" THEN 8
10
320 FORX=1000TO1012
330 READA:POKEX,A:NEXTX
340 POKE360,3:POKE361,232
350 DATA 52,16,142,0,1
360 DATA 189,167,211,53
370 DATA 16,126,130,113
380 POKE1003,15:PALETTE7,P:CLS8:
WIDTH40:ATTR3,7:POKE65496,0
390 C$="PASSWORD1":D$="PASSWORD2
"
400 LOCATE5,3:PRINT"Enter Passwo
rd Details.";:LOCATE8,8:PRINT"Na
me: ";:LOCATE8,13:PRINT"Password
: ";
410 ATTR6,7:LOCATE14,8:PRINT"";:
LINE INPUT"";A$
420 ATTR6,7:LOCATE18,13:PRINT"";
:LINE INPUT"";B$
430 IF A$<>C$ GOTO 450 ELSE 440
440 IF B$<>D$ GOTO 450 ELSE 460
450 ATTR3,7:LOCATE5,20:PRINT"Inv
alid Password Attempt.":FORX=1TO
100:NEXTX:GOTO380
460 ATTR3,7:POKE1003,0:FORX=1TO1
00:NEXTX
470 POKE65497,0:HSCREEN2:HCLS3
480 PALETTE1,42:HCOLOR1:HLINE(50
,40)-(280,150),PSET,BF:HCOLOR2:H
LINE(52,43)-(278,147),PSET,B:HCO
LOR14:HPRINT(9,7),"Please Select
 From Menu.":HPRINT(12,10),"1.
Protect Disk.":HPRINT(12,13),"2.
  Restore Disk.":HPRINT(12,16),"
3.  End Program."
490 POKE65496,0
500 A$=INKEY$:IF A$="" THEN 500
510 IF A$="1" THEN 540 ELSE 520
520 IF A$="2" THEN 730 ELSE 530
530 IF A$="3" THEN 810
540 WIDTH40:CLS:LOCATE5,10:INPUT
"ENTER DATE(DD/MM/YY)";DA$
550 DSKI$ 0,17,2,A$,B$
560 IF MID$(A$,68,1)=CHR$(&HC9)
THEN 590
570 IF MID$(A$,68,1)=CHR$(&HFF)
THEN 590
580 WIDTH40:CLS:LOCATE5,10:PRINT
"GRAN 68 IN USE...CONTINUE(Y/N)"
:INPUT Z$:IF Z$="Y" THEN 590 ELS
E 470
590 MID$(A$,68,1)=CHR$(&HC9)
600 DSKO$ 0,17,2,A$,B$
610 DSKO$ 0,34,11,A$,B$
620 FOR DS=3 TO 9
630 DSKI$ 0,17,DS,A$,B$
```

Continued on p18

NOISEWORKS

By Steve Youngberry

ARTICLE

IF YOU ARE A beginner when it comes to programming then this idea will probably be a nightmare.

If, on the other hand, you just want an end result, then you might find this of some interest.

Those of you who know anything about M/L after reading this might make a noise that goes like this: YECH!!!.

The idea is to have M/L sounds instead of Basic <POPS> and <SQUEAKS> in your Basic program.

If you don't know how to program in M/L then you (like me) are up the creek.

Well not any more! While this may be a long winded process, it does have the desired end result - a Basic program with M/L sounds and music.

First you need a couple of programs. The first program is "Music+" from July '86 and the second is "Data Generator" like the one from CoCo June '85.

The first thing to do is to RUN "Music+" and discover your new sound, be it effect or music.

After you are satisfied, you then save it onto disk or tape.

Exit "Music+" and RUN "Data Generator" and convert your M/L file from MUSIC+ into Basic DATA statments.

Easy so far.

In theory you could have up to 10 different sounds by using the DEFUSR function. I ran into some trouble with a lot and you also end up with a page of data statements VERY quickly.

Next comes the merging - disk users will have no problem. Tape users will have to use a merge program or print out the listing and retype it in after you have CLOADed your program (or look up January's 1986 edition of CoCo in the Hints and Tips sections of the magazine. There's a hint there relating to merging two basic programs if you only have a tape. -ed.)

Don't forget to RENUMber your program accordingly. You can use more than one sound, but you must not overlap the M/L subroutines.

Work this out by subtracting the address of the first byte from the address of the end byte of the first M/L subroutine and then add the result +1 to the start, end, and exec address of the second subroutine.

Now somewhere along the way you may notice the nice little tune becomes an unbearable noise (I don't know why ...).

Someone somewhere will be able to tell us why (I hope they do).

Also, the manual tells us to save Basic's stack pointer before using the USR fuction and restore it before going back to Basic if the M/L subroutine is more than 30 bytes long.

I couldn't find the stack if it bit me nor could I save or restore it.

Maybe this is where I'm going wrong. Anyway, it is workable and the sound effects are much better than Basic's.

Below is a little help.

```
10 CLEAR19712,951:CLEAR20285,573
```

This clears enough memory for two subroutines. This line generaly caused am OM error so I deleted it (what ever works).

```
20 FOR A=19712 TO 20663:READ Q$:
POKE  I,VAL("&H"+Q$):NEXT  A
30 FOR A=20664 TO 21237:READ Q$:
POKE I,VAL("&H"+Q$):NEXT A
```

With "Music+" the start and exec addresses are always the same, so in line 30 the data is POKEd into addresses 951+1 higher than the first routine. (951=len of the first routine +1)

```
40 DEFUSR0=20100:DEFUSR1=21051
```

Continued on p18

WORKING DATA STRUCTURES *Pt 3*

By John Redmond ARTICLE

FOR THE LAST TWO months, we have been discussing some aspects of writing a full screen editor in Forth, and how the correct data structures can make the program more elegant and efficient. This time, we look at an ASCII file as a data structure.

The simplest text file is simply an unstructured stream of characters, with an End Of File mark, which might be 0 (ASCII null) as in Unix, or 26 (control-Z) as in CP/M. Within such a file, there will probably be control characters, such as 13 (for carriage return), 10 (line feed) and 9 (tab). These will affect the way in which the text will be displayed on a screen or a piece of paper.

The EDTASM+ line editor saves a text, to tape or disk, in just this way. But, when it loads such a file, it holds it in memory as a linked series of dimensioned strings (see 'Data Structures in Forth' a few months ago).

I mention this, only to make the point that there are several ways of handling text, but we will not pursue this sort of structure further at this stage.

In some ways, Forth seems to show its age. The text file which it uses is the very simplest, and it relates directly to the way all data are saved on disk. Regardless of data type, this is done in 1k blocks and, for text, the 1k is organized into 16 lines of 64 characters.

There are NO control characters, like carriage return. When the text is displayed on screen, during an edit, the lines are normally placed in the middle of the screen, with work and prompt space above and/or below. If a line is altered, it is immediately updated, in its proper place, on the screen.

If a line is to be printed, it has first to be found in the text buffer. I would prefer to avoid the intricacies of Forth virtual memory management at this stage, but suffice it to say that a disk screen (say number 27) can be got into memory by 27 BLOCK. BLOCK does all the work of finding a buffer for the screen and reading it from the disk into the buffer.

When it finishes the job, it returns on the stack the address of the memory buffer that has been allocated. For our present purposes, this value can be saved in a variable:

```
VARIABLE BUFFSTART

27 BLOCK BUFFSTART !
```

Now, logically, this buffer is considered as a 16 x 64 array of characters (bytes). The first 64-character line (line 0) will start at the address given by BUFFSTART @.

The addresses of other lines are best given by a special word:

```
: LINESTART ( #line--add)

  64 * ( get array offset)

  BUFFSTART @ ( base add)

  + ; ( line add)
```

LINESTART expects a line number on the stack and returns the address of that line. Very easy, but note that LINESTART does no range checking (what if the line number is negative, or greater than 15?). The checking should be done somewhere else in the program, where the line number is thought up in the first place.

Once we have the address of the first character, we can do things with the line. Because we are going to use the value many times in the program, it is best to define ...

```
64 CONSTANT /LINE
```

This name is pronounced 'perline'. What perline? Characters, of course! If we want to type line 4 (say), we code ...

```
4 LINESTART /LINE TYPE
```

But maybe there are not 64 characters on the time - perhaps only 41? Then we will code ...

```
4 LINESTART -TRAILING TYPE
```

Remember that -TRAILING (pronounced 'not-trailing') trims trailing spaces from ASCII strings. It is very useful for speeding up writing to the high-resolution screen (why waste time writing spaces?).

Remember, though, that a full-screen editor must put the lines in their proper places on the screen. If we have no spaces wasted at the top, line 4 of the text will appear in the position for logical line 4 on the screen, and this is where the cursor must be placed before starting the print.

This is done with TAB.

```
  0 4 TAB
```

... will do the job, but it is better to define a special word

```
: .LINE  ( #line)

 0 OVER ( get line no)

 TAB ( locate cursor)

 LINESTART ( obvious)

 -TRAILING ( trim line)

 TYPE ; ( display it)
```

As the editor is implemented, there is space taken at the top of the screen, for messages etc., so that this definition will not put the line in the correct position. Assume that four lines are used at the top and we have another short

definition ...

```
: +TOP  4 + ;
```

Now, all we have to do is insert +TOP before TAB in the above definition and we have the desired display.

But there are other things the editor needs to do with the text strings. We need to move lines, and to insert and remove them. Want to move line 4 to line 11?

```
4 LINESTART 11 LINESTART /LINE

CMOVE
```

Feel the power? Want to remove a line? Before we do that, we must recognize that the buffer has a fixed length - and has an end.

```
1024 CONSTANT /BUFFER

: BUFFEND  ( #buff--add)

  BUFFSTART @ /BUFFER + ;
```

Note that we have defined BUFFEND in terms of the value contained in BUFFSTART at the time BUFFEND is invoked. It is NOT a constant but, as far as the calling program is concerned, it is indistinguishable from a constant, like /BUFFER. Each is only a single word, but how they do their jobs is no business of the rest of the program. Such information hiding makes software maintenance very easy and orderly.

The job of removing a line comes down to 1. moving a block of text from the start of the next line down to the start of the current line; and 2. filling the last line with blanks. The main problem in part 1. is determining the size of the text block to be moved.

This is given by the difference between BUFFEND and the start of that next line (think about it!). Part 2 is easier: the address of the last line is given by BUFFEND /LINE - and the standard Forth word, BLANK, is used to fill it with the correct number of ASCII spaces. Therefore

```
: XLINE  ( #line)

  DUP 1+ ( #next line)

  LINESTART ( its add)

  DUP ( extra copy)

  ROT ( # this line)

  LINESTART ( its add)

  BUFFEND

  ROT ( add next line)

  - ( block length)

  CMOVE ( do the move)

  BUFFEND  /LINE - ( last)

  /LINE BLANK ;
```

It's worth saying here that the definition of XLINE is about as long as any word definition should get. There are two Forth dirty words (DUP and ROT) in the definition. These are part of the tremendous power that a Forth programmer has over her/his data stack.

The values are there, but they are so abstract that they don't even have names. No other language has such mystical powers, but things can get out of hand. By using the dirty words OVER, DUP, SWAP and ROT excessively, it is possible to write code that is just as obscure as C.

This is a lesson that every budding Forth programmer has to learn personally. If there are lots of dirty words, the definitions are too long and/or the code is badly factored. (Ironically, the textbooks say that C definitions should be short and well-named, but noone seems to have heeded that advice.)

While I'm on my high horse, I point out the lack of punctuation in Forth. It doesn't need those awful commas, colons, semicolons and brackets. It gets by with the humble space, assisted by good factoring and the much-maligned postfix notation.

Is it just possible that those other languages are wrong after all?

Well, that's all for a while. I feel that gentle reader is due for a spell without me, but I also feel that s/he needs a spell WITH Forth. As always, I'm available at 23 Mirool St., West Ryde, NSW 2114 or (better) (02)-85-3751 after 7pm.

Come Forth!

From p16

```
640 IF DS=9 THEN MID$(B$,120,LEN
(DA$))=DA$
650 DSKO$ 0,34,DS+9,A$,B$
660 NEXT DS
670 MID$(B$,120,8)=DA$
680 FOR DS=1 TO 18
690 A$="*LOCKED*"
700 DSKO$ 0,17,DS,A$,B$
710 NEXT DS
720 GOTO 800
730 DSKI$ 0,34,18,A$,B$
740 DA$=MID$(B$,120,8)
750 WIDTH40:CLS:LOCATE6,8:PRINT
"DATE LAST LOCKED WAS "DA$:LOCAT
E8,11:INPUT"CONTINUE RESTORE(Y/N
)";A$:IF A$<>"Y" THEN 470
760 FOR DS=2 TO 9
770 DSKI$ 0,34,DS+9,A$,B$
780 DSKO$ 0,17,DS,A$,B$
790 NEXT DS
800 GOTO 470
810 WIDTH32:RGB:POKE65496,0:END
```

From p16

This tell the computer where to find the routine when you call it.

```
50-???? program
100 P=USR0(0)
```

EXEC the first routine by typing ...

```
150 O=USR1(0)
```

EXEC the second routine ...

```
200 DATA,,,,,,,,,
```

To explain the DEFUSR and USR functions refer to the manual - I can fumble though them, but I'm not really sure what goes on.

If anyone can simplify this method or gets lost trying to follow it, drop me a line or a phone call will eventually get to me.

My address is PO Box 244, Tara, 4421

AUTO EXEC AND PASSWORD

By John Baker 32K ECB + EDTASM+ OR SOME SORT OF ASSEM

IN JUNE'87 COCO, there was a program written by BILL SNOW on how to Auto-execute Basic programs and password protect them.

I found a way to combine the AUTOLOAD program into my assembly program to enable Basic and Machine language programs to auto-execute.

In listing 1, you will see in line 10 that the program begins with ORG 386 - this is a RAM HOOK. There are a few of these in the Basic operating system.

They are changeable so that you can direct Basic to the beginning of your program with a JMP START.

There are other ways but they are more complicated.

In line 40 my logo is placed on the screen while loading - this can be changed for your own needs.

Line 80 is the beginning of the password routine. The password may be changed for yourself by changing the word in line 370 to your own and by changing the LDX #n value in line 230 to the length of your password.

Line 680 is the start of the auto load and run program from Bill Snow.

Listing 2 must begin with lines 5 + 6. All of your programs must have lines 5 + 6 and begin at line 10.

This is so nobody may escape the password subroutine.

You may not even want password protection in your programs.

If so, simply delete lines 70 to 600 and delete lines 5+6 of your Basic programs.

You may not want the front logo either - if so, simply delete lines 30 to 60.

First, save your assembled program straight after your Basic program. Leave the record and play keys depressed during it.

Oh! And one other point - when saving the assembly language part make sure it has the same name as the Basic program.

Any trouble? Write to
John Baker
63 Tucker Rd.
Moorabbin, 3189

The Listing:

```
00005 *AUTO EXEC BY JOHN BAKER
   IF JUST AN ASSEMBLY PROGRAM
   JUST PUT IN ORG 386 JMP
   START
00006 *THEN BEGIN YOUR PROGRAM
   WITH THE LABEL START,YOU
   CAN ALSO USE THE PASSWORD
   ROUTINE
00010  ORG 386
00020  JMP START
00030  ORG $400+64
00040  FCC / JOHN BAKER SOFTWAR
E  /
00050  FCC /      PRESENTS /
00060  FCC /   DELTA ATTACK!  /
00070  ORG 4000
00080 START1 LDA #$7E
00090  STA 386
00100  LDA #$0F
00110  STA 387
00120  LDA# $E4
00130  STA 388
00140  LDY #PSW
00150  LDA $5FFF
00160  CMPA #76
00170  LBE QLOCK
00180  LDB #9
00190  LDX #1024
00200  LBSR BLACK
00210  LBSR PRINT
00220 START2 LDY #PASS
00230  LDX #5
00240  BRA START4
00250 START3 LEAY +1,Y
00260 START4 JSR [$A000]
00270  BEQ START4
00280  CMPA ,Y
00290  LBNE LOCK
00300  LEAX -1,X
00310  BEQ STAT
00320  BRA START3
00321 LOCKL BSR BLACK
00322  LDY #PSW
00323  LDB #9
00324  LDX #1024
00325  LBSR PRINT
00330  LDA #76
00340  STA $5FFF
00350 LOCK1 JMP LOCK1
00360 PSW FCC /PASSWORD?/
00370 PASS FCC /LOGON/
00380 STAT LDA #57
00390  STA 386
00400  STA 387
00410  STA 388
00420  LDA #67
00430  STA 3999
00440  JMP 44661
00450 BLACK PSHS A,B,X,Y
00460  LDX #$400
00470  LDA #128
00480 BLAC1 STA ,X+
00490  CMPX #1024+512
00500  BEQ BLAC2
00510  BRA BLAC1
00520 BLAC2 PULS Y,X,B,A
00530  RTS
00540 PRINT LDA ,Y+
00550  CMPA #64
00560  BLO FIX
00570 PRIN1 STA ,X+
00580  DECB
00590  BNE PRINT
00600  RTS
00610 FIX CMPA #32
00620  BEQ FIX1
00630  ADDA #64
00640  BRA PRIN1
00650 FIX1 LDA #128
00660  BRA PRIN1
00670  ORG 3584
00680 START LDA 57
00690  STA 386
00700  STA 387
00710  STA 388
00720  INC 104
00730  CLR 465
00740  JSR 42572
00750  LDX 25
00760  LEAX -1,X
00770  CLR ,X+
00780  JSR 44313
00790  JSR 42876
00800  LDX 25
00810 LOOP STX 126
00820  LDD 126
00830  INC A
00840  JSR 44087
00850  JSR 42763
00860  FCB 16
00870  FCB 38
00880  FCB 150
00890  FCB 211
00900  LDA 124
00910  FCB 16
```

Continued on p24

CONVERSATION

by Martin Eade

GAME
CoCo 3/CoCo2 Modifiable

CONVERSATION IS mainly for fun but is a good demonstration of several Basic techniques.

It has set responses to around 75 words and is based on a program found in "Better Basic" by B. Reffin Smith & L. Watts.

Any sentence can be entered and it will either randomly generate a sentence or print a response to your last one.

It incorporates the triple speed poke, so it may not work on some systems without modifications.

So if your CoCo can't stand the triple speed POKE, then get rid of it.

The Listing:

```
1 GOTO4
2 CSAVE"CONVERSA"
3 SAVE"305B:3":END'1
4 'CONVERSATION MK.II-A NEW
 VERSION OF THE CONVERSATION
 PROGRAM IN THE BOOK BETTER BASI
C.MK II BY MARTIN EADE WITH SOME
 HELP FROM CRISTIAN SOUTHALL
99 CLEAR2000
100 CLS
110 DIM V$(10),N$(10),A$(10),T$(
10),S$(10),M$(68),Q$(68),C(68)
150 GOSUB1000
210 LINEINPUT"HELLO,WHATS YOUR N
AME?";D$
220 PRINT:PRINT"TALK TO ME ";D$
230 LINEINPUT I$
240 IF I$=""THEN230
245 POKE65497,0
250 IF I$="GET LOST"THEN 910:IF
I$="BYE"THEN925
310 REPLY=RND(8)
320 IF REPLY<6THEN490 ELSE 600
340 POKE65496,0:PRINT:PRINTR$:PR
INT
380 R$=""
400 T=0
410 FORK=1TO68
420 T=T+C(K)
430 NEXTK
440 IFT<12THEN460
450 FOR K=1TO68:C(K)=0:NEXTK
460 T=0
470 GOTO230
490 FOR PHRASE=1TO68
500 L1=LEN(Q$(PHRASE))
510 L2=LEN(I$)
520 FOR TEST=1TO L2
530 IF MID$(I$,TEST,L1)=Q$(PHRAS
E)THEN550
540 NEXT TEST:NEXT PHRASE:IF R$=
""THEN600ELSE340
550 IF C(PHRASE)>0THEN540
560 C(PHRASE)=C(PHRASE)+1
570 R$=R$+M$(PHRASE):R$=R$+".":N
EXT TEST
580 GOTO340
600 '
610 E=RND(10):F=RND(10):G=RND(10
):H=RND(10):L=RND(10)
620 ON E GOTO 700,720,740,760,78
0,800,830,850,870,890
700 R$="WHAT DO YOU THINK ABOUT
"+N$(H)+"?"
710 GOTO340
720 R$=S$(L)+" "+D$+" YOU DON'T
THINK ALL HUMANS ARE "+A$(G)+" D
O YOU?":GOTO340
740 R$="I'VE HEARD THAT YOU ARE
ARE SOME KIND OF "+A$(G)+" "+T$(
H)+" "+D$:GOTO340
760 R$=S$(L)+" "+D$+" ,I THINK Y
OU ARE JUST AS "+A$(G)+" AS THE
OTHER PEOPLE I'VE TALKED TO":GOT
O340
780 R$="I AM FEELING "+A$(G)+" N
OW"
790 GOTO340
800 PRINT:PRINT"SSSHHHH.....I AM
 THINKING ....":R$="LETS "+V$(F)
+" "+N$(H)+" I THINK "+N$(H)+" I
S "+A$(G):GOTO340
830 R$="TELL ME ABOUT "+N$(H)+",
"+D$
840 GOTO340
850 R$="DO YOU THINK I AM "+A$(G
)+","+D$+"?"
860 GOTO340
870 R$="LETS "+V$(F)+" SOMETHING
 ELSE MORE "+A$(G):GOTO340
890 R$="GUESS WHAT I AM THINKING
 "+D$
900 GOTO340
910 POKE65496,0:PRINT"SAME TO YO
U!!!":END
925 POKE65496,0:PRINT"HAVE A NIC
E DAY,";D$;"!!!":END
1000 FOR I=1TO 66:READ Q$(I):NEX
TI
1001 DATA WHO ARE," WHAT ",?,MEA
N,WHY,"ME ","I "," IT ",TALK," N
O ",?," ARE "," MY ",YES,?,THINK
,CLEVER,RUDE,THANK,OFF,THEY,?,UN
DERSTAND," NOT "," IS ",TO,?,KNO
W,NAME,COLOUR,NUMBER,"GOING OUT
WITH",VIC,64,APPLE,AMIGA,"SUCK",
MALE,FEMALE
1002 DATA" AGE ",SMASH,BASH UP,M
ICROCHIP,CPU,VDU,SAD,HAPPY,CLEVE
R,DUMB,INTELLIGENT,RUDE,SEX,EAT,
DRINK,BURP,SCHOOL,SPECIES,BOYFRI
END,GIRLFRIEND,MUSIC,PETER,VIOLE
NCE,DROP,SAW,SHUT UP,YOU
1100 FORI=1TO66:READ M$(I):NEXTI
1105 DATA I AM ONLY A COMPUTER
1110 DATA I DON'T KNOW,WHAT WAS
THE QUESTION?,GA GA GOO GOO,"WEL
L,WHY NOT?",OH,I HAVE A SERIOUS
MENTAL PROBLEM,WILL YOU SHUT UP,
YOU'RE A JOCKSTRAP,YOU TELL ME,I
 AM A NON-WALKING HEAP OF JUNK,M
Y-MY-MY
1111 DATAI DISAGREE,I HATE YOUR
GUTS,HIT ME,PICKY-PICKY,THANKS,Y
OU'VE SEEN NOTHING YET,"DON'T ME
NTION IT (MY FEE FOR    THAT WAS
 $50,000,000)",YOU SUCK,I DON'T
CARE,WHAT A STUPID QUESTION,YOU'
VE GOT A LOW I.Q.,RUBBISH,WHAT M
AKES YOU SO SURE?,GO AWAY
1112 DATA GET LOST,KNOWLEDGE IS
A PROBLEM FOR ME,MY NAME IS COCO
,MY FAVOURITE COLOUR IS VOMIT GR
EEN
1114 DATA MY PHONE NUMBER I AM N
OT TELLING,PERVERT!,"VIC-20'S AR
E OBSOLETE!",HE'S GOT SO MUCH ME
MORY!OHHH!,HE'S GOT SO MUCH MEMO
RY!OHHH!,HE'S GOT SO MUCH MEMORY
!OHHH!,I DO NOT SUCK,I AM FEMALE
,I AM FEMALE,ASK MARTIN,GO BACK
TO YOUR CAGE
1115 DATA I'M GETTING OUT OF HER
E,6809E AND PROUD OF IT!,"NICE,I
SN'T IT?",YOU'RE A SAD CASE,LUCK
Y YOU,?????,ARE YOU REALLY AS DU
MB AS YOU    LOOK?,YOU ARE THE ON
LY LIVING ORGANISMWITH AN IQ BEL
OW ZERO!,I'M SUPER INTELLIGENT
1116 DATA DIRTY ORGANISM,ARE YOU
 A NYMPHO OR SOMMAT?
1117 DATA I'M ON A DIET,"HIC!I'M
 AN ALCOFROLIC!",PIG!,YOU SWORE!
,ARE YOU INSULTING ME AGAIN?,MIN
D YOUR OWN BUSINESS!,I DON'T HAV
E A GIRLFRIEND(I'M FEMALE),LA LA
 LA DO RE MI FA ETC ETC,"UGH! YO
```

Continued on p24

ILLUSION

By Nicholas Fuller

EDUCATION GRAPHICS
COCO 3

WRITTEN FOR THE CoCo 3, "Illusions"'s aim is to demonstrate common optical illusions. Most programs for the CoCo in education seem to be aimed at primary school students - well, what about high school and uni students?

I hope this program goes a long way in achieving it's aim for uni and high school students.

This program has all the instructions for the different illusions. I intend to write more programs for high school students in the near future.

This is a really interesting subject which deals with our "perception" and just some of the illusions include, The Necker Cube, Ponze Illusion, Size and Perspective.

Have fun!

The Listing:

```
0 GOTO10
1 '********** ILLUSIONS ********
   ***** NICHOLAS FULLER *******
3 SAVE"316B:3":END'6
10 POKE65497,0
20 HBUFF1,621:HBUFF2,621
30 HSCREEN2:FORT=1TO10:PALETTE0,
6:A=A+7:HCIRCLE(160,96),A,1:NEXT
T
40 PALETTE12,63:HCOLOR12,2
50 HPRINT(13,2),"Optical Illusio
ns":PALETTE11,50:HCOLOR11,3:HLIN
E(90,167)-(240,175),PSET,BF:HCOL
OR1,2:HPRINT(12,21),"By Nicholas
 FULLER"
60 FORT=1TO61:SOUNDRND(255),1:PA
LETTE11,RND(63):PALETTE12,RND(63
):FORX=1TO50:NEXTX:PALETTE1,RND(
63):NEXTT:PALETTE12,63:PALETTE1,
0
70 HDRAW"BM260,140;C1;D40R40U40L
40E20R40G20E20D40G20"
80 HGET(10,5)-(30,55),1:HDRAW"BM
30,110;C12G10R5D30R10U30R5H10":H
PAINT(30,115),12,12
90 HGET(20,110)-(40,160),2
100 FORVV=1TO30:HLINE(260,20)-(3
10,100),PSET,B:HPAINT(270,30),10
,12:AX=RND(50):AY=RND(80):HCIRCL
E(260+AX,20+AY),5,12:HPAINT(260+
AX,20+AY),RND(10),12:NEXTVV
110 FORT=1 TO300:NEXTT
120 FOREE=1TO6
130 HPUT(20,110)-(40,160),1
140 HPUT(20,70)-(40,120),2:FORT=
1TO300:NEXTT:HPUT(20,70)-(40,120
),1
150 HPUT(20,20)-(40,70),2:FORT=1
TO300:NEXTT:HPUT(20,20)-(40,70),
1
160 HPUT(20,110)-(40,160),2:FORT
=1TO300:NEXTT:HPUT(20,110)-(40,1
60),1
170 NEXTEE
180 HPUT(20,110)-(40,160),2
190 PALETTE3,5:PALETTE4,7:PALETT
E5,8:PALETTE6,16:PALETTE7,20:PAL
ETTE8,25:PALETTE9,28:PALETTE10,4
3
200 HPAINT(160,96),3,1:HPAINT(17
0,96),4,1:HPAINT(177,96),5,1:HPA
INT(187,96),6,1:HPAINT(197,96),7
,1:HPAINT(207,96),8,1:HPAINT(217
,96),9,1:HPAINT(227,96),10,1
210 HPAINT(265,145),5,1:HPAINT(3
15,145),7,1:HPAINT(285,135),10,1
220 FORR=1TO10
230 FORXX=1TO25:READA:READB:PALE
TTEA,B:FORV=1TO10:NEXTV:PALETTE1
,A:NEXTXX
240 RESTORE:NEXTR
250 HPRINT(3,10),"press any key"
260 A$=INKEY$:IF A$=""THEN260
270 WIDTH40:PALETTERGB
280 ATTR3,2,B:PRINT"****";:ATTR4
,6:PRINT"WELCOME TO";:ATTR2,1,B,
U:PRINT"ILLUSIONS";:ATTR3,2,B:PR
INT"***";
290 PRINT"   ":ATTR2,5,B:PRINT"I
";:ATTR3,3:PRINT"llusions is a p
rogram designed to educate and s
how some of the common";:ATTR3,3
,U:PRINT" optical illusions";:AT
TR3,3:PRINT"that exist."
300 PRINT"   ";:PRINT"All of the
 illusions are examples of our p
hyscology and it is quite amazin
g how our brain,eyes react in di
fferent situations and ways.Your
 brain is a very complex piece o
f material which performs many s
trange things subconsiencely"
310 PRINT"  ";:PRINT"This progra
m would be suited to anyone in h
igh-school or uni":INPUT"PRESS <
ENTER>";A$
320 LPOKE516158,RND(63)
330 WIDTH40:CLS:ATTR3,2,B:PRINT"
   ";:PRINT"MENU":ATTR2,5:PRINT"
1)QUIT":PRINT"2)NECKER AND OTHER
 CUBES":PRINT"3)MULLER-LYER AND
OTHER SPECIAL LINES":PRINT"4)SIZ
E AND PERSPECTIVE":PRINT"5)COLOR
 AND THE EYE":PRINT"6)THE BRAIN"
:PRINT"7)SAVE PROGRAM"
340 INPUTA:ON A GOSUB350,470,660
,1100,1510,2200,400
350 CLS:INPUT"DO YOU WISH TO EXI
T TO BASIC";A$:IF A$="N"THEN330E
LSE360
360 CLS1:ATTR4,5,U,B:PRINT"Bibli
ography";:ATTR3,2:PRINT" ":PRINT
"The ENCYCLOPAEDIA BRITTANICA-VO
L 9 pages 240-247":PRINT"An Intr
oduction to PHYSCOLOGY":PRINT"Wo
rld book ENCLCYCLOPAEDIA PGS 664
-668":PRINT"The human body "
370 PRINT"PRESS ANY KEY"
380 EXEC44539
390 CLS6:POKE65496,0:PRINT"Writt
en & produced by nicholas fuller
 PH(02) 5161518 ***JULY 1987***"
:ATTR3,3:PRINT"another solar sof
tware program":FORT=1TO2000:NEXT
T:CLS:END
400 CLS:POKE65496,0:FORT=1TO15:P
LAY"T255O1V30DADADADADADACACACFF
FFAAAAV-":NEXTT:ATTR0,4:PRINT"IN
STUCTIONS TO SAVE":ATTR3,3:PRINT
"Place a blank tape in recorder
or disk in appropriate place and
 hold down play/record if applic
able"
410 INPUT"DISK or TAPE SYSTEM";A
A$:IF AA$="TAPE" OR AA$="T" THEN
 420 ELSE 440
420 CLS:INPUT"READY";KK$:IF KK$=
"Y"THEN430 ELSE 420
430 CSAVE"ILLUSION":GOTO10
440 CLS:INPUT"READY";JJ$:IF JJ$=
"Y" THEN 450 ELSE 440
450 SAVE"******":GOTO10
460 DATA 3,5,4,7,5,8,6,16,7,20,8
,25,9,28,10,43,3,5,4,7
470 CLS:ATTR3,3:PRINT"   ":PRINT
"THE NECKER CUBE":PRINT"  ":PRIN
T"The NECKER cube is one of the
most famous optical illusions.Wh
en ever we see something whether
 it be things that you see everd
ay or unfamiliar items, your bra
in constantly checks it and t
480 PRINT"arrage it into a group
 of things or in pairs or compar
e it to something we are all rea
dy familiar with.This is the the
ory behind the NECKER cube and o
```

```
ther similar drawings.As you loo
k at the cube(s) you will see th
at your brain scans over";
490 PRINT"it serching out all th
e different possibilities.As a r
esult you see the cube 'flip' fo
rward and backward.The effective
ness of the puzzle seems to work
 better with younger rather than
 older people"
500 PRINT"PRESS ANY KEY TO VIEW
THE";:ATTR5,4,B,U:PRINT"Necker C
ube"
510 A$=INKEY$:IF A$=""THEN510
520 HSCREEN4:PALETTE3,62:PALETTE
1,0:HCLS3:HCOLOR1,2:HPRINT(10,0)
,"The Necker Cube":HPRINT(10,20)
,"ANY KEY"
530 HLINE(100,30)-(250,20),PSET:
HLINE-(250,100),PSET:HLINE-(100,
115),PSET:HLINE-(100,30),PSET
540 HLINE-(100,115),PSET:HLINE-(
185,106),PSET:HLINE-(185,70),PSE
T:HLINE-(345,55),PSET:HLINE-(345
,135),PSET:HLINE-(185,150),PSET:
HLINE-(185,70),PSET
550 HLINE-(100,30),PSET:HLINE(25
0,20)-(345,55),PSET:HLINE(345,13
5)-(250,100),PSET:HLINE(185,150)
-(100,115),PSET
560 PALETTE2,21:HPAINT(150,60),2
,1:HPAINT(200,60),2,1:HPAINT(200
,90),2,1
570 A$=INKEY$:IF A$=""THEN570
580 HCLS3:XZ$="C1R100D50L100U50"
:HDRAW"BM100,20"+XZ$:HDRAW"BM50,
50"+XZ$:HDRAW"BM50,100"+XZ$:HDRA
W"BM100,130"+XZ$:HLINE-(50,100),
PSET:HLINE(150,100)-(200,130),PS
ET:HLINE(200,180)-(150,150),PSET
590 HLINE(100,180)-(50,150),PSET
:HLINE(100,20)-(50,50),PSET:HLIN
E(100,70)-(50,100),PSET:HLINE(20
0,70)-(150,100),PSET:HLINE(200,2
0)-(150,50),PSET:HCIRCLE(100,100
),6,1:HPAINT(100,99),1,1:HPAINT(
100,101),1,1
600 HLINE(300,0)-(300,192),PSET:
HPRINT(5,0),"Another Necker Cube
":HPRINT(41,4),"We continually e
xtract paterns from the ":HPRINT
(38,5),"things we see,trying to
match them with":HPRINT(38,6),"s
omething meaniful.Stare at the d
ot in the"
610 HPRINT(38,7),"centre of the
two cubes to establish for":HPRI
NT(38,8),"yourself the flucuatin
g nature of percept-":HPRINT(39,
9),"ion.Your brain performs all
sorts of transf-":HPRINT(38,10),
"sformations seeing different pa
tterns "
620 HPRINT(38,11),"in the cube":
HPRINT(38,13),"In the previous c
ube notice how the shaded":HPRIN
T(38,14),"area could be either a
t the front or back"
630 HPRINT(38,20),"PRESS ANY KEY
"
640 A$=INKEY$:IF A$=""THEN 640
650 PALETTERGB:GOTO330
660 'LINES
670 HSCREEN4:PALETTERGB
680 PALETTE3,62:HCLS3:HDRAW"BM20
0,50;C2H10F10G10E10R300E10G10F10
":HDRAW"BM190,80;C2E10G10F10H10R
300H10F10G10"
690 PALETTE4,4:HCOLOR4,2:HPRINT(
10,2),"MULLER-LYER":HPRINT(10,30
),"Press any key"
700 A$=INKEY$:IF A$=""THEN700
710 WIDTH40:PALETTERGB:ATTR3,3:P
RINT"MULLER-LYER ILLUSION":PRINT
:PRINT"This illusion is based on
 the Gesalt principles of conver
gence and diversion:the lines at
 the side seem to lead the eye e
ither inwards or outwards to cre
ate a false impression of";
720 PRINT"length":PRINT:PRINT:PR
INT:PRINT"PRESS ANY KEY"
730 A$=INKEY$:IF A$=""THEN730
740 HSCREEN4:HPRINT(10,0),"LINES
"
750 HLINE(200,70)-(270,70),PSET:
HLINE(200,130)-(270,130),PSET:HL
INE(200,30)-(130,160),PSET:HLINE
(265,30)-(340,160),PSET:HPRINT(2
0,30),"ANY KEY"
760 A$=INKEY$:IF A$=""THEN760
770 HCLS:HPRINT(14,2),"The Pogge
ndorff Illusion"
780 HLINE(100,50)-(400,170),PSET
,B:HLINE(145,60)-(185,160),PSET,
B:HLINE(255,60)-(295,160),PSET,B
:HLINE(310,60)-(350,160),PSET,B
790 HLINE(105,55)-(145,90),PSET:
HLINE(185,125)-(240,165),PSET:HL
INE(245,50)-(255,60),PSET:HLINE(
295,105)-(310,120),PSET:HLINE(35
0,160)-(360,170),PSET
800 HPRINT(10,30),"PRESS ANY KEY
"
810 A$=INKEY$:IF A$=""THEN810
820 HCLS:HPRINT(10,2),"THE HERIN
G ILLUSION":HLINE(60,50)-(450,15
0),PSET,B:FORXX=60TO450STEP10:HL
INE(245,50)-(XX,100),PSET:HLINE-
(245,150),PSET:NEXTXX:HLINE(60,9
0)-(450,90),PSET:HLINE(60,110)-(
450,110),PSET
830 HPRINT(10,27),"PRESS ANY KEY
"
840 EXEC44539
850 WIDTH40:PRINT"POGGENDORFF &
HERING ILLUSIONS":PRINT"The prev
ious illusions all rely on what
is called perception,that is how
 our mind interprets what we see
.In the Poggendorff illusion the
 proximity of the two rectangles
 appear to bend the lines";
860 PRINT".The effectivness of t
he illusion depends on the steep
ness of the line":PRINT"The Heri
ng illusion --in this illusion t
he various lines seem to lead ou
r eyes in the wrong direction to
 create an illusion that they ar
e bent"
870 PRINT"In Lines the lines are
 of equal length.This illusion m
akes use of Convergence,an examp
le of this is a railway track in
 the distance We see that the sl
eepers are smaller the further t
hey get but realize that they ar
e of equal length"
880 PRINT"press any key"
890 EXEC44539
900 HSCREEN4
910 HCLS:HPRINT(10,2),"LINES CON
TINUED":HLINE(100,50)-(300,160),
PSET,B:FORCI=1TO10:R=R+10:HCIRCL
E(200,105),R,1:NEXTCI:HLINE(130,
75)-(270,135),PSET,B
920 HPRINT(0,30),"PRESS ANY KEY"
930 EXEC44539
940 WIDTH40:PRINT"ILLUSIONS":PRI
NT"In that illusion the concentr
ic circles create the false impr
ession of a bent square"
950 PRINT"PRESS ANY KEY"
960 EXEC44539
970 HSCREEN4
980 HCLS:HLINE(100,50)-(300,150)
,PSET:HLINE(100,50)-(300,150),PS
ET,B:HLINE(140,50)-(300,130),PSE
T:HLINE(180,50)-(300,110),PSET:H
LINE(220,50)-(300,90),PSET
990 HLINE(100,70)-(260,150),PSET
:HLINE(100,90)-(220,150),PSET:HL
INE(100,110)-(180,150),PSET
1000 HPRINT(10,0),"ZOLLNER'S ILL
USION":HPRINT(10,30),"PRESS ANY
KEY"
1010 FORL=1TO19:A=A+10:B=B+5:C=4
3+B:D=B+57:HLINE(100+A,C)-(100+A
,D),PSET:NEXTL:FORLL=1TO10:Z=Z+1
0:X=X+5:Q=83+X:Y=X+98:HLINE(100+
Z,Q)-(100+Z,Y),PSET:NEXTLL
1020 FORLM=1TO10:S=S+10:F=F+5:G=
50+F:H=60+F:HLINE(190+S,G)-(190+
S,H),PSET:NEXTLM
1030 FORQQ=1TO7:R=R+10:T=T+5:U=5
0+T:V=60+T:HLINE(110+R,U)-(130+R
,U),PSET:NEXTQQ
1040 FORQQ=1TO15:AS=AS+10:TR=TR+
5:PP=45+TR:HLINE(120+AS,PP)-(140
+AS,PP),PSET:NEXTQQ
1050 FORQQ=1TO15:AD=AD+10:TE=TE+
5:PO=70+TE:HLINE(89+AD,PO)-(112+
AD,PO),PSET:NEXTQQ:FORQQ=1TO7:AW
=AW+10:AA=AA+5:WA=110+AA:HLINE(9
0+AW,WA)-(110+AW,WA),PSET:NEXTQQ
1060 EXEC44539
1070 WIDTH40:PRINT"ZOLLNER ILLUS
ION":PRINT"The cross hatching of
 the smaller lines distrubes the
 parallel lines to create an imp
ression of them not being so":PR
INT"press any key"
1080 EXEC44539
1090 PALETTERGB:GOTO 330
1100 WIDTH40:ATTR2,4,B:PRINT"Siz
e and perspective":ATTR2,5:PRINT
"The following illusion(koning)i
s a perfect example of size and
perspective and how they can mis
lead the brain"
1110 PRINT"PRESS ANY KEY"
1120 A$=INKEY$:IFA$=""THEN1120
1130 HSCREEN4:PALETTERGB:HPRINT(
```

```
10,0),"KONING-Size & perspective
"
1140 HCIRCLE(150,90),5,1:HCIRCLE
(170,91),8,1:HCIRCLE(195,93),12,
1:HCIRCLE(228,95),16,1:HCIRCLE(2
70,98),22,1:HCIRCLE(312,95),16,1
:HCIRCLE(345,93),12,1:HCIRCLE(37
0,91),8,1:HCIRCLE(390,90),5,1:HL
INE(50,50)-(500,150),PSET,B:HLIN
E(40,45)-(510,155),PSET,B
1150 HPRINT(10,20),"PRESS ANY KE
Y":HPAINT(42,52),7,4:PALETTE3,RN
D(63)
1160 A$=INKEY$:IFA$=""THEN1150
1170 WIDTH40:ATTR3,2:PRINT"SIZE
AND PERSPECTIVE":ATTR2,5:PRINT"T
OPS OF THE CIRCLES IN THE PREVIO
US DIAGRAM WERE ALL IN A STRAIGH
T LINE":PRINT"THE NEXT TWO DIAGR
AMS WILL ALSO DEMONSTRATE THE IL
LUSION OF SIZE THE CIRCLES WILL
APPEAR DIFFERENT SIZES -WHEN";
1180 PRINT" ARE IN ACTUAL FACT E
QUAL":PRINT"PRESS ANY KEY"
1190 A$=INKEY$:IF A$=""THEN1190
1200 HSCREEN2:PALETTE3,24:HCLS3:
HPRINT(8,0),"SIZE & PONZO ILLUSI
ONS":HCIRCLE(160,100),8,2:HPAINT
(160,100),2,2:HCIRCLE(80,100),8,
2:HPAINT(80,100),2,2:HCIRCLE(80,
100),12,2:HCIRCLE(160,100),30,2
1210 HPRINT(10,30),"PRESS ANY KE
Y"
1220 EXEC44539
1230 HCLS3:HCOLOR4,1:HPRINT(10,0
),"THE PONZO ILLUSION":HCOLOR2,2
:HLINE(200,100)-(60,40),PSET:HLI
NE(200,100)-(60,140),PSET:HCIRCL
E(170,99),11,2:HCIRCLE(90,99),11
,2
1240 HPRINT(10,30),"PRESS ANY KE
Y"
1250 EXEC44539
1260 HSCREEN4:PALETTERGB:PALETTE
3,52:HCLS3
1270 HPRINT(10,0),"MORE ILLUSION
S":HPRINT(10,30),"PRESS ANY KEY
TO CONTINUE"
1280 HDRAW"BM150,70;C2D90R140E50
L20G50R20L20U15E50D15U15L20G50R2
0L20U15E50D15U15L20G50R20L20U15E
50D15U15L20G50R20L20U15E50D15U15
L20G50R20L20U15E50D15U15L20G50R2
0L20U15E50D15U15L20G50R20L20E50R
140D90"
1290 HPAINT(160,80),6,2:HPAINT(2
60,32),6,2
1300 EXEC44539
1310 WIDTH40:ATTR3,2,U,B:PRINT"I
LLUSIONS":ATTR2,3:PRINT"The prev
ious illusion was similar to the
 Necker Cube in the fact that it
 ll continually 'flopped' betwee
n a normal staircase or an upsid
e down one"
1320 INPUT"DO YOU WANT TO SEE TH
AT AGAIN";DD$:IF DD$="Y" THEN 12
60 ELSE 1330
1330 HSCREEN2:HCOLOR4,2:HPRINT(1
0,0),"**ILLUSIONS**":HPRINT(10,3
0),"PRESS ANY KEY":HLINE(80,50)-
(40,150),PSET:HLINE-(240,150),PS
ET:HLINE-(280,50),PSET:HLINE-(80
,50),PSET:HLINE(60,150)-(150,50)
,PSET:HLINE-(230,150),PSET:HLINE
(150,50)-(80,150),PSET
1340 EXEC44539
1350 WIDTH40:PRINT"ILLUSIONS":PR
INT"The line intersecting the tw
o lines creates the impression t
hat the line to the right is lon
ger than the one to the left":PR
INT"In the following illusions r
ead tham carfully .Did you read
them correctly?"
1360 FORT=1TO3000:NEXTT
1370 HSCREEN2:HCLS7:HPRINT(10,3)
,"MORE ILLUSIONS":HPRINT(10,30),
"PRESS ANY KEY":HDRAW"BM80,50;C2
G70R140H70F70R140H70G70":HPAINT(
80,56),5,2:HPAINT(220,56),5,2
1380 HPRINT(7,10),"PARIS":HPRINT
(6,12),"IN THE":HPRINT(4,14),"TH
E SPRING":HPRINT(25,10),"ONCE":H
PRINT(24,12),"UPON A":HPRINT(24,
14),"A TIME"
1390 EXEC44539
1400 WIDTH40:ATTR3,2:PRINT"**ILL
USIONS**":PRINT"nicholas fuller
1987":PRINT"Look very carefully
at the next illusion .Do you see
 3 or 5 cubes?":INPUT"DO YOU WIS
H TO SEE THAT AGAIN";DD$:IF DD$=
"Y" THEN 1350
1410 HSCREEN4:PALETTE2,33:HCLS2:
HCOLOR3,2:HDRAW"BM50,170;C1E25F2
5G25H25E25F25E25F25G25H25E25F25E
25F25G25H25":HDRAW"BM50,170;C1U2
5E25D25R3U25F22D25H22F22R3U25E22
D25R3U25D25F22U25H22F22R3E22D25G
22E22R3U25F22D25H22"
1420 HDRAW"BM78,120;C1E22R3F22R3
E22R3F22U25H22D25L3U25G22D25L3U2
5H22D25L3U25G22D25U25E22R3E22F22
R3":HPAINT(80,120),1,1:HPAINT(13
0,120),1,1:HPAINT(130,150),1,1:H
PAINT(72,150),1,1:HPAINT(180,150
),1,1:HPAINT(120,80),1,1:HPAINT(
0,0),3,1
1430 HPAINT(100,179),3,1:HPAINT(
150,179),3,1:HCOLOR2,2:HPRINT(10
,0),"CUBES 3 or 5?":HPRINT(30,0)
,"PRESS ANY KEY"
1440 EXEC44539
1450 WIDTH40:PRINT"ILLUSIONS":IN
PUT"DO YOU WISH TO SEE THAT AGAI
N";DD$:IF DD$="Y" THEN 1410
1460 PALETTERGB:GOTO 330
1470 DATA 3,7,4,5,5,7,6,8,7,16,8
,20,9,25,10,20,9,25,10,28,3,43,4
,5,0,33,3,0,4,10,5,20,6,33,7,44,
8,33,9,44,10,33
1480 DATA3,5,4,7,5,5,6,7,7,8,8,1
6,9,20,10,25,3,28,4,43,3,43,4,5,
5,7,6,5,7,7,8,8,9,16,10,20,3,25,
4,28,3,28,4,43,5,5,6,7,7,5,8,7,9
,8,10,16,3,20,4,25,3,25,4,28,5,4
3,6,5,7,7,8,5,9,7,10,8,3,16,4,20
1490 DATA 3,20,4,25,5,28,6,43,7,
5,8,7,9,5,10,7,3,8,4,16,3,16,4,2
0,5,25,6,28,7,43,8,5,9,7,10,5,3,
7,4,8,3,8,4,16,5,20,6,25,7,28,8,
43,9,5,10,7,3,5,4,7
1500 DATA 3,7,4,8,5,16,6,20,7,25
,8,28,9,43,10,5,3,7,4,5
1510 WIDTH40:PALETTERGB:ATTR4,3:
PRINT"COLOUR AND THE EYE":PRINT"
Color and the eye plays a very i
mportant aspect in all optical i
llusions.The eye focuses light o
n the Retina via the lens of the
 eye and the optical nerve trans
mits this information to the b
1520 PRINT"Color, like shapes ca
n also play tricks on the brain
as seen in the following illusio
n.Some people suffer from color
blindness ie do not recognize ce
rtain colors or get them mixed u
p.8 males in every 100 is color
blind compared with 1 for ";
1530 PRINT"every females":PRINT"
In the folloing illusions does t
he rectangle surrounded by black
 appear brighter than the one su
rrounded by white?":PRINT"PRESS
ANY KEY"
1540 EXEC44539
1550 HSCREEN2:PALETTE3,28:PALETT
E4,0:PALETTE5,63:HCLS9:HCOLOR5,2
:HPRINT(10,0),"BRIGHTNESS AND CO
LOR":HPRINT(10,30),"PRESS ANY KE
Y"
1560 HCOLOR3,2:HLINE(50,50)-(210
,150),PSET,BF:HCOLOR4,3:HLINE(60
,55)-(135,145),PSET,BF:HCOLOR5,4
:HLINE(135,55)-(205,145),PSET,BF
:HCOLOR3,2:HLINE(80,80)-(115,120
),PSET,BF:HLINE(150,80)-(185,120
),PSET,BF:EXEC44539
1570 WIDTH40:PALETTERGB:INPUT"DO
 YOU WANT TO SEE THAT AGAIN";KK$
:IF KK$="Y" THEN 1550
1580 PRINT"ILLUSIONS":PRINT"In t
he next illusion stare at the do
t for about 30 seconds and then
look at the white sheet,you shou
ld see the complementary colors
as an after image
1590 PRINT"If it goes away try b
linking to restore the image. pr
ess any key"
1600 EXEC44539
1610 HSCREEN2:PALETTE7,57:HCLS7:
HCOLOR3,2:HPRINT(10,0),"Compleme
ntary colors":HPRINT(10,30),"PRE
SS ANY KEY"
1620 PALETTE2,18:PALETTE3,9:PALE
TTE4,38:PALETTE5,50:HCIRCLE(160,
80),40,3,1,.64,.875:HLINE(160,80
)-(132,53),PSET:HLINE(160,80)-(1
89,53),PSET:HCIRCLE(160,90),40,2
,1,.13,.38:HCOLOR2,3:HLINE(160,9
0)-(130,120),PSET:HLINE(160,90)-
(190,120),PSET
1630 HCOLOR4,3:HCIRCLE(160,85),4
0,4,1,.4,.62:HLINE(154,85)-(130,
60),PSET:HLINE(154,85)-(130,110)
,PSET:HCOLOR5,4:HCIRCLE(160,85),
40,5,1,.9,.1:HLINE(165,85)-(195,
60),PSET:HLINE(166,85)-(195,110)
,PSET:PALETTE9,0:HCIRCLE(160,85)
,5,9:HPAINT(160,85),9,9
1640 HPAINT(160,70),3,3:HPAINT(1
```

```
60,100),2,2:HPAINT(180,90),5,5:H
PAINT(133,90),4,4
1650 EXEC44539
1660 WIDTH80:PALETTERGB:ATTR4,4,
B:PRINT"COLOR BLINDNESS":ATTR3,3
:PRINT"Color blindness or DALTON
ISM is a condition where the per
son has trouble or can not ident
ify different colours":PRINT"Man
y colour blind people see only t
wo of the colors of the";
1670 PRINT" rainbow, they see ye
llows and blues but confuse reds
 and greens.Most color blind peo
ple don't realize they are color
 blind and learn to use the comm
on names for particular colors.
Very few people who are color bl
ind see only black and white";
1680 PRINT"That is they posses a
chromatic vision (the other bein
g called dichromatic vision).Eig
ht percent men are color blind,
compared with 0.5 percent women.
 Many animals are naturally colo
r blind and do not see colors as
 we do"
1690 PRINT"Color Blindness is in
herited and it can be tested by
the HARDY-RAND-RITLER and ISHIRO
MA tests which consist of a figu
re hidden in a jumble of differn
et colored dots to test a person
s ability to recognize colors"
1700 PRINT"PRESS ANY KEY"
1710 EXEC44539
1720 HSCREEN2:PALETTE1,0:HCOLOR1
,0:PALETTE2,57:HCLS2:HPRINT(10,0
),"COLOR BLINDNESS":HPRINT(10,30
),"PRESS ANY KEY":HCIRCLE(100,50
),10,3:HPAINT(100,50),3,3:HCIRCL
E(130,50),10,9:HPAINT(130,50),9,
9
1730 HLINE(70,30)-(160,130),PSET
,B:PALETTE5,50:PALETTE4,13:HCIRC
LE(100,90),10,5:HPAINT(100,90),5
,5:HCIRCLE(130,90),10,4:HPAINT(1
30,90),4,4
1740 HLINE(180,20)-(320,180),PSE
T,B:HPRINT(23,5),"Color blind":H
PRINT(23,6),"people would find "
:HPRINT(23,7),"the red & green":
HPRINT(23,8),"as the same color"
:HPRINT(23,10),"The yellow & blu
e":HPRINT(23,11),"would be seen"
1750 HPRINT(23,12),"as equal col
ors"
1760 EXEC44539
1770 WIDTH40:PALETTERGB:ATTR3,3,
B:PRINT"          ";:PRINT"ILLUSIO
NS****glossary of terms":ATTR3,2
:PRINT"ACHROMATIC COLORS-colors
black ,grey & white":PRINT"AFTER
IMAGE-sensory experience that re
mains after stimulus is taken aw
ay ie train of colored images
1780 PRINT"BLIND SPOT-unsensitiv
e area of the eye where the reti
na joins the optic nerve":PRINT"
CHROMATIC COLORS-all other color
s than black,white or grey":PRIN
T"DARK ADAPTION-the adaption to
different intensities of light i
e after coming from outside";
1790 PRINT"into a room it usuall
y looks darker than it is as you
r eyes change(with use of rods)t
o the light. Another example is
when you walk into a dark room i
t takes a few minutes for your e
yes to adjust"
1800 PRINT"DEPTH PERCEPTION-The
perception of distance in solid
objects from front and back or f
rom odinary objects":PRINT"DICHR
OMATISM-color blindness in reds,
greens or yellow or blue.PRESS A
NY KEY"
1801 EXEC44539
1802 CLS:ATTR3,3,B:PRINT"       "
;:PRINT"ILLUSIONS**glossary of t
erms":ATTR3,2
1810 PRINT"DISTANCE CUES-The thi
ngs we use to see how far things
 are away from us ie light & sha
dow,perspective and relative mov
ement":PRINT"GESALT PSYCOLOGY-A
system of psychological theory b
ased on perception which uses pa
tterns etc"
1820 PRINT"ILLUSION-A misreprese
ntation of relationships among a
ny stimuli so that what you see
is not actually as it is physica
l":PRINT"PERCEPTION-The process
of which becoming aware of relat
ionships/patterns etc":PRINT"STE
REOSCOPIC VISION-Binocular ";
1821 PRINT"perception of depth &
 distance caused by overlapping
of eye fields":PRINT"TRICHROMATI
SM-Normal color vision":PRINT"PR
ESS ANY KEY"
1822 EXEC44539
1830 GOTO 330
2200 CLS:PRINT"THE BRAIN":PRINT"
  ";:PRINT"The brain is the cont
rol centre of the brain and is d
ivided into two halves, the righ
t & left hemispheres":PRINT"The
right half controls the left sid
e of the body and the left hand
side ofthe brain controls the
2210 PRINT"side of the body.A li
nk between the two halves ensure
s that each side know what the o
ther is doing.The brain has many
 differnet areas which do a vari
ety of things"
2220 PRINT"Visual infomation is
stored at the back,sounds at sid
es and speech at the front.Memor
y & judgement are carried out in
 the front of the brain and sens
es like heartbeat,breathing etc
also are at the front."
2230 INPUT"PRESS ENTER";DD$
2240 CLS:PRINT"    ";:PRINT"THE B
RAIN/EYE":PRINT"The eye is very
similar to a camera and focuses
light via the lens and the iris
which controls the amount of lig
ht entering the eye,onto the ret
ina":PRINT"The image is upside d
own and it is coverted into el
2250 PRINT" impulses to the brai
n where it is unjumbled":PRINT"T
he optic nerve carries these sig
nals to the brain.":PRINT"PRESS
ANY KEY"
2260 EXEC44539
2270 CLS:PRINT"CONCLUSION":PRINT
"Hopefully you have seen that wh
at you see and think you see are
 two different things altogether
.Your brain is a marvellous piec
e of equipment so be really care
ful of it**you can't get a new o
ne (yet!)."
2280 PRINT"I also hope you learn
t about perception and in the pr
ocess enjoyed the illusions demo
nstrating my points.":PRINT"REME
MBER--SEEING IS NOT BELIEVING!!!
":PRINT"PRESS ANY KEY"
2290 EXEC44539
2295 GOTO 330
3000 GOTO 3000
```

■

From p19

```
MIT,SPEW,CHUCK","DON'T GET VIOLE
NT....PLEASE!"
1118 DATA MY GIME CHIP IS SETTLE
D-DON'T DROP ME,I NEED GLASSES,S
NIFF SNIFF-YOU HATE ME DON'T YOU
?,"WHAT ARE YOU?"
1119 FORI=1TO10:READ N$(I):NEXTI
1120 DATA FOOTBALL,BALLROOM DANC
ING,SCHOOL,THE WEATHER,AIDS,SEX,
THE WEATHER,FOOD,GEOGRAPHY,ME
1121 FORI=1TO10:READ V$(I):NEXTI
1122 DATA THINK ABOUT,TALK ABOUT
,DISCUSS,CONTEMPLATE,REFLECT ON,
MEDITATE ON,COGITATE,PONDER,CERE
BRATE,CONSIDER
1123 FORI=1TO10:READ A$(I):NEXTI
1124 DATA STUPID,CLEVER,INTELLIG
ENT,WISE,WITTY,DENSE,THICK AS A
BRICK,RUDE,NEUROTIC,DERANGED
1125 FORI=1TO10:READ S$(I),T$(I)
:NEXT I
1126 DATA GOOD HEAVENS,BORE,"WEL
L,",TOAD,LET'S SEE,JERK,LISTEN,G
ENIUS,LOOK,DUMBO,"UMMM...",MORON
,NOV,PARASITE,REALLY,PRODIGY,OH
NO,MONSTER,MY DEAR FRIEND,COMPUT
ER FREAK
1127 RETURN
```

From p20

```
00920  FCB 39
00930  FCB 150
00940  FCB 20
00950  BPL LOOP
00960  STX 27
00970  JSR 42985
00980  JSR 44321
00990  JSR 44271
01000  JMP 44446
01010  END
```

Your number is up with...

YAHTZEE

by Frank Woodward

32K ECB
GAME

by Frank Woodward

THE OBJECT OF YAHTZEE is to obtain the higest score. Each player may roll the dice up to 3 times before having to enter their score in one of the 13 sections.

On each roll you may keep any of the five dice to produce the desired combination.

Press <enter> for yes, or "N" and <enter> for no. If you make a mistake press "R" and <enter>.

Each player must complete the full 13 sections and is invited to enter their player number 1-4 to enter their score on their score card.

Sections 1-6 score the total of all dice. A bonus of 35 is awarded if the total of these sections is greater than 62.

A "yahtzee" is 5 dice all the same and a bonus of 100 is awarded for every yahtzee after the first in any one game.

The other sections are self explanitory.

Have fun, and may the best one win!

The Listing:

```
0 GOTO20
1 '******** YAHTZEE ************
2 '******* BRENDON PUDNEY ******
3 SAVE"307A:3":END'1
20 WIDTH32
40 CLS2
42 FOR X=1TO6:A=RND(6):NEXT X
45 A$=CHR$(230)
47 PRINT@128,STRING$(32,A$);
49 PRINT@204,"YAHTZEE";:PRINT@26
3,"BY FRANK WOODWARD";
51 PRINT@320,STRING$(32,A$);
53 PRINT@390,"INSTRUCTIONS Y/N";
:INPUT"";B$
54 CLS1
55 IF B$="Y" THEN GOTO 57 ELSE 8
0
57 PRINT@1,"YAHTZEE A GAME FOR 1
-4 PLAYERS"
58 PRINT@64,"THE OBJECT OF YA
HTZEE IS TO OBTAIN THE HIGHEST
SCORE FOR ONE OR MORE GAMES."
59 PRINT"EACH PLAYER MAY ROLL
 THE DICE UP TO 3 TIMES BEFORE H
AVING TO    ENTER THEIR SCORE IN
TO ONE OF THE 13 SECTIONS, ON EA
CH ROLL  THEY MAY KEEP ANY OF TH
E 5 DICE TO   PRODUCE THE DESIRE
D COMBINATION.";:INPUT"   PRESS
<ENTER>";A$
60 CLS:PRINT"DURING THIS SEQUENC
E YOU ARE    INVITED TO KEEP OR
DELETE ANY   OF THE 5 DICE, PRES
S <ENTER> FORYES.   <N> AND <ENT
ER> FOR NO.  IF YOU MAKE A MISTA
KE PRESS <R> AND <ENTER>. "
61 INPUT"EACH PLAYER MUST COMPLE
TE THE   FULL 13 SECTIONS AND IS
 INVITED TO ENTER THEIR NUMBER (
1-4)  TO ENTER THEIR TOTAL SCORE
 ON THE  SCORE CARD.  PRESS <ENT
ER>";A$
62 PRINT"SCORING."
63 CLS:PRINT"SECTIONS 1-6 SCORE
THE TOTAL OF ALL DICE OF THAT VA
LUE,IF THE   TOTAL OF 1-6 IS GRE
ATER THAN 62 THEN A BONUS OF 35
IS AWARDED.  A LARGE STRAIGHT IS
 ALL 5 DICE  IN SEQUENCE ,A SMAL
L STRAIGHT   IS 4 DICE. A  YAHTZ
EE IS 5 DICE ALL THE SAME."
64 PRINT"A FULL HOUSE IS 3 DICE
OF ONE   VALUE AND 2 OF ANOTHER.
A BONUS  OF 100 IS AWARDED FOR
 EVERY    YAHTZEE AFTER THE FIRS
T IN ANY  ONE GAME":INPUT"PRESS
<ENTER>";A$:GOTO80
70 RESTORE:CLS1
75 TP=0:CL=0:TA=0:TB=0:TC=0:TD=0
:TE=0:TF=0:TG=0:TH=0:TI=0:TJ=0:T
K=0:TL=0:TM=0:TN=0
76 QA=0:QB=0:QC=0:QD=0:QE=0:QF=0
:QG=0:QH=0:QI=0:QJ=0:QK=0:QL=0:Q
M=0
80 A$=CHR$(204)
100 B$=CHR$(207)
120 C$=CHR$(192)
140 D$=CHR$(202)
160 E$=CHR$(200)
180 DATA 0,6,12,18,24,32,38,44,5
0,56,64,70,76,82,88
200 FOR X=1TO15
220 READF
240 H$=STRING$(5,B$)
300 PRINT@F,C$
320 PRINT@F+1,H$
340 PRINT@F+6,C$
360 PRINT@F+7,C$
380 NEXTX
400 PRINT@96,C$
420 PRINT@97,STRING$(5,A$)
440 PRINT@102,C$
460 PRINT@103,STRING$(5,A$)
480 PRINT@108,C$
500 PRINT@109,STRING$(5,A$)
520 PRINT@114,C$
540 PRINT@115,STRING$(5,A$)
560 PRINT@120,C$
580 PRINT@121,STRING$(5,A$)
600 PRINT@126,E$
620 PRINT@127,C$
640 RESET(60,0):RESET(60,1):RESE
T(60,2):RESET(60,3):RESET(60,4):
RESET(60,5):RESET(60,6)
645 M$=CHR$(172)
650 PRINT@160,STRING$(32,M$)
655 TT=1:GOSUB 2400
660 P=2
675 PRINT@129,"PRESS <ENTER> TO
ROLL DICE";:INPUT"";A$
680 IF P=2 GOTO 700 OR ELSE IF P
=3 GOTO 1640 ELSE 2000
700 DATA 3,15,27,39,51,60,40,50,
60,3,15,27,39,51,60,40,50,60,3,1
5,27,39,51,60,40,50,60,3,15,27,3
9,51,60,40,50,60,3,15,27,39,51,6
0,40,50,60,3,15,27,39,51,60,40,5
0,60
710 DATA 3,15,27,39,51,60,40,50,
60,3,15,27,39,51,60,40,50,60,3,1
5,27,39,51,60,40,50,60,3,15,27,3
9,51,60,40,50,60,3,15,27,39,51,6
0,40,50,60,3,15,27,39,51,60,40,5
0,60,3,15,27,39,51,60,40,50,99
720 FOR X=1TO5
740 NEXT X
760 READ D
780 IF D=60 GOTO 800 ELSE 820
800 GOTO 1260
820 FOR T=1TO150
840 NEXT T
860 PLAY"O3L64;C;L32;D;L64;C"
880 A=RND(6)
890 GOSUB 3200
900 IF A=1 THEN GOTO 1220 ELSE 9
20
920 IF A=2 THEN GOTO 1180 ELSE 9
40
940 IF A=3 THEN GOTO 1140 ELSE 9
60
960 IF A=4 THEN GOTO 1100 ELSE 9
```

```
80
980 IF A=5 THEN GOTO 1060 ELSE 1
000
1000 IF A=6 THEN GOTO 1020
1020 RESET(D,1):RESET(D+3,1):RES
ET(D+3,5):RESET(D+6,5):RESET(D,5
):RESET(D+6,1)
1040 GOTO680
1060 RESET(D,1):RESET(D+6,1):RES
ET(D+3,3):RESET(D,5):RESET(D+6,5
)
1080 GOTO 680
1100 RESET(D,1):RESET(D+6,1):RES
ET(D,5):RESET(D+6,5)
1120 GOTO 680
1140 RESET(D,1):RESET(D+3,3):RES
ET(D+6,5)
1160 GOTO 680
1180 RESET(D,1):RESET(D+6,5)
1200 GOTO 680
1220 RESET(D+3,3)
1240 GOTO 680
1260 GOTO 1280
1280 READ L
1285 GOSUB 3602
1286 YA=0:IF TV=10 GOSUB 2070
1290 IF L=99 GOTO 2720
1300 IF L=60 GOTO 1360
1305 IF TV=10 GOTO 1410
1310 PRINT@128,STRING$(32,.)
1312 M$=CHR$(188)
1315 PRINT@160,STRING$(32,M$);
1320 PRINT@135,"ANOTHER ROLL Y/N
";
1340 INPUT"";A$
1345 M$=CHR$(128+16*(3-1)+12)
1347 PRINT@160,STRING$(32,M$);
1350 IF A$<>"N" THEN GOTO 1420 E
LSE 1410
1360 PRINT@128," LAST ROLL ";:IN
PUT"WHICH AREA 1-13";A$
1370 IF A$=""GOTO 1360
1373 IF VAL(A$)<1 GOTO 1360
1375 IF VAL(A$)>13 GOTO 1360
1380 P=3:GOSUB 2740
1400 GOTO 675
1410 IF L=40 GOTO 1415
1412 IF L=50 GOTO 1417
1415 READ L
1417 READ L
1418 P=3 :GOTO 2720
1420 PRINT@128,STRING$(32,.)
1425 AA=0:AB=0:AC=0:AD=0:AE=0
1430 PRINT@135,"KEEP DICE 1 Y/N"
;
1440 INPUT"";C$:IF C$="N" THEN A
A=2:ELSE IF C$="R" THEN GOTO 142
5
1480 PRINT@135,"KEEP DICE 2 Y/N"
;
1500 INPUT"";D$:IF D$="N" THEN A
B=2:ELSE IF D$="R" THEN GOTO 142
5
1520 PRINT@135,"KEEP DICE 3 Y/N"
;
1540 INPUT"";E$:IF E$="N" THEN A
C=2:ELSE IF E$="R" THEN GOTO 142
5
1560 PRINT@135,"KEEP DICE 4 Y/N"
;
1580 INPUT"";F$:IF F$="N" THEN A
D=2:ELSE IF F$="R" THEN GOTO1425
1600 PRINT@135,"KEEP DICE 5 Y/N"
;
1620 INPUT"";G$:IF G$="N" THEN A
E=2:ELSE IF G$="R" THEN GOTO1425
1622 IF AA=2 THEN GOSUB1640 ELSE
 1624
1624 IF AB=2 THEN GOSUB 1740
1626 IF AC=2 THEN GOSUB 1760
1628 IF AD=2 THEN GOSUB 1780
1630 IF AE=2 THEN GOTO 1800
1635 GOTO 1840
1640 FOR A=2TO9:FOR B=1TO5:SET(A
,B,5):NEXT B,A:IF P=3 GOTO 1740
ELSE RETURN
1740 FOR C=14TO21:FOR D=1TO5:SET
(C,D,5):NEXT D,C:IF P=3 GOTO 176
0 ELSE RETURN
1760 FOR E=26TO33:FOR F=1TO5:SET
(E,F,5):NEXTF,E:IF P=3 GOTO 1780
 ELSE RETURN
1780 FOR G=38TO45:FOR H=1TO5:SET
(G,H,5):NEXT H,G:IF P=3 GOTO 180
0 ELSE RETURN
1800 FOR I=50TO57:FOR J=1TO5:SET
(I,J,5):NEXT J,I:IF P=3 GOTO 182
0 ELSE 1840
1820 GOTO 660
1840 PRINT@129,"       ROLLING TH
E DICE"
1860 P=1
1880 IF C$="N" THEN D=3:IF D=3 G
OTO 820 ELSE 1900
1900 IF D$="N" THEN D=15:IF D=15
 GOTO 820 ELSE 1920
1920 IF E$="N" THEN D=27:IF D=27
 GOTO 820 ELSE 1940
1940 IF F$="N" THEN D=39:IF D=39
 GOTO 820 ELSE 1960
1960 IF G$="N" THEN D=51:IF D=51
 GOTO 820 ELSE 2000
1980 IF P=1 GOTO 1260
2000 IF D=3 GOTO 1900 ELSE 2020
2020 IF D=15 GOTO 1920 ELSE 2040
2040 IF D=27 GOTO 1940 ELSE2060
2060 IF D=39 GOTO 1960 ELSE1260
2070 M$=CHR$(236)
2072 CL=CL+100
2075 PRINT@128,"          ITS A Y
AHTZEE"
2080 PRINT@160,STRING$(32,M$);
2085 PLAY"T2L64O4CDEFGACAGFEDCCD
EFGABAGFEDCCDEFGABAGFEDCCDEFGABA
GFEDCCDEFGAB"
2090 FOR T=1TO500:NEXT T:RETURN
2200 PRINT@129,"          WRONG S
ECTION      "
2220 SOUND 10,10:FOR T=1TO400:NE
XT T
2240 GOTO2720
2400 N$=CHR$(133)
2410 R$=CHR$(147)
2420 O$=CHR$(138)
2430 S$=CHR$(130)
2500 PRINT@192,"1"N$" ONES "O$:P
RINT@201,TA:PRINT@204,O$" 7"N$"3
 OF A KIND "O$:PRINT@221,TG
2520 PRINT@224,"2"N$" TWOS "O$:P
RINT@233,TB:PRINT@236,O$" 8"N$"4
 OF A KIND "O$:PRINT@253,TH
2540 PRINT@256,"3"N$"THREES"O$:P
RINT@265,TC:PRINT@268,O$" 9"N$"
FULL HOUSE "O$:PRINT@285,TI
2560 PRINT@288,"4"N$"FOURS "O$:P
RINT@297,TD:PRINT@300,O$"10"N$"S
M STRAIGHT "O$:PRINT@317,TJ
2580 PRINT@320,"5"N$"FIVES "O$:P
RINT@329,TE:PRINT@332,O$"11"N$"L
G STRAIGHT "O$:PRINT@349,TK
2600 PRINT@352,"6"N$"SIXES "O$:P
RINT@361,TF:PRINT@364,O$"12"N$"
CHANCE      "O$:PRINT@381,TC
2660 R$=CHR$(128+16*(1-1)+3)
2670 PRINT@385,N$"BONUS "O$:PRIN
T@393,TN:PRINT@396,O$"13"N$" YAH
TZEE     "O$:PRINT@413,TM
2680 PRINT@416,STRING$(8,R$):PRI
NT@424,R$:PRINT@425,STRING$(3,R$
):PRINT@428,S$:PRINT@429,STRING$
(13,R$):PRINT@442,S$:PRINT@443,S
TRING$(5,R$)
2690 PRINT@448," BONUS ":PRINT@4
60,O$" GRAND TOTAL "O$:PRINT@475
,TQ
2695 PRINT@481,"TARGET";
2700 RETURN
2720 PRINT@128," ENTER WHICH SEC
TION 1-13";:INPUT"";A$
2725 IF A$=""GOTO 2720
2730 IF VAL(A$)<1 GOTO 2720
2735 IF VAL(A$)>13 GOTO 2720
2740 TT=2
2760 IF A$="1" THEN U=202:IF U=2
02 AND QA=1 THEN GOTO 2200:ELSE
GOSUB 3300:TA=TZ:QA=1
2780 IF A$="2" THEN U=234:IF U=2
34 AND QB=1 GOTO 2200:ELSE GOSUB
 3300:TB=TZ:QB=1
2800 IF A$="3"THEN U=266:IF U=26
6 AND QC=1 GOTO 2200:ELSEGOSUB 3
300:TC=TZ:QC=1
2820 IF A$="4" THEN U=298:IF U=2
98 AND QD=1 GOTO 2200:ELSEGOSUB
3300:TD=TZ:QD=1
2840 IF A$="5" THEN U=330:IF U=3
30 AND QE=1 GOTO 2200:ELSEGOSUB
3300:TE=TZ:QE=1
2860 IF A$="6" THEN U=362:IF U=3
62 AND QF=1 GOTO 2200:ELSEGOSUB
3300:TF=TZ:QF=1
2880 IF A$="7" THEN U=222:IF U=2
22 AND QG=1 GOTO 2200:ELSEGOSUB
3600:TG=TZ:QG=1
2900 IF A$="8" THEN U=254:IF U=2
54 AND QH=1 GOTO 2200:ELSEGOSUB
3600:TH=TZ:QH=1
2920 IF A$="9" THEN U=286:IF U=2
86 AND QI=1 GOTO 2200:ELSEGOSUB
3600:TI=TZ:QI=1
2940 IF A$="10" THEN U=318:IF U=
318 AND QJ=1 GOTO 2200:ELSEGOSUB
 3600:TJ=TZ:QJ=1
2960 IF A$="11" THEN U=350:IFU=3
50 AND QK=1 GOTO 2200:ELSEGOSUB
3600:TK=TZ:QK=1
2970 IF A$="12" THEN U=382:IFU=3
82 AND QL=1 GOTO 2200:ELSEGOSUB
3600:TL=TZ:QL=1
2980 IF A$="13" THEN U=414:IFU=4
14 AND QM=1 GOTO 2200:ELSEGOSUB
3600:TM=TZ:QM=1
2994 IF L=99 GOTO 3140
2995 GOTO 680
3010 TZ=VAL(B$)
3020 IF LEN(B$)=2 THEN GOTO 3040
```

```
 ELSE 3080
3040 C$=LEFT$(B$,1):D$=RIGHT$(B$
,1)
3060 C=VAL(C$):D=VAL(D$):GOTO 31
00
3080 B=VAL(B$)
3085 IF B=0 GOTO 3110
3090 PRINT@U,CHR$(B+48);:RETURN
3100 PRINT@U,CHR$(C+48)+CHR$(D+4
8);:RETURN
3110 PRINT@U,CHR$(42)+CHR$(42);:
RETURN
3140 TX=TA+TB+TC+TD+TE+TF:IF TX>
=63 THEN TN=35:ELSE IF TX<63 THE
N TN=0
3160 XX=TX+TN+TG+TH+TI+TJ+TK+TL+
TM
3165 IF TN=35 GOTO 3170 ELSE 318
0
3170 PRINT@394,CHR$(51)+CHR$(53)
;
3180 PRINT@475,XX;:GOTO 3800
3200 IF D=3 GOTO 3210:ELSE IF D=
15 GOTO 3220:ELSE IF D=27 GOTO 3
230:ELSE IF D=39 GOTO 3240:ELSE
IF D=51 GOTO 3250
3210 DA=A:RETURN
3220 DB=A:RETURN
3230 DC=A:RETURN
3240 DD=A:RETURN
3250 DE=A:RETURN
3300 IF U=202 THEN V=1:ELSE IF U
=234 THEN V=2:ELSE IF U=266 THEN
 V=3:ELSE IF U=298 THEN V=4:ELSE
 IF U=330 THEN V=5:ELSE IF U=362
 THEN V=6
3320 IF DA<>V THEN DA=0
3330 IF DB<>V THEN DB=0
3340 IF DC<>V THEN DC=0
3350 IF DD<>V THEN DD=0
3360 IF DE<>V THEN DE=0
3365 T=V*3
3370 TZ=DA+DB+DC+DD+DE
3375 TP=TZ-T+TP
3377 PRINT@456,TP;
3380 IF TZ=0 GOTO 3440
3400 IF TZ>9 GOTO 3460
3420 PRINT@U,CHR$(TZ+48);:RETURN
3440 PRINT@U,CHR$(42)+CHR$(42);:
RETURN
3460 BB$=STR$(TZ)
3480 C$=MID$(BB$,2,1):D$=RIGHT$(
BB$,1)
3500 C=VAL(C$):D=VAL(D$)
3520 PRINT@U,CHR$(C+48)+CHR$(D+4
8);:RETURN
3600 TV=0:TZ=DA+DB+DC+DD+DE:GOTO
 3604
3602 TV=0:YA=1
3604 CA=0:CB=0:CC=0:CD=0:CE=0:CF
=0:CG=0:CH=0:CI=0:CJ=0
3605 IF DA=DB THEN TV=TV+1:IF DA
=DB THEN CA=1
3606 IF DA=DC THEN TV=TV+1:IF DA
=DC THEN CB=1
3607 IF DA=DD THEN TV=TV+1:IF DA
=DD THEN CC=1
3608 IF DA=DE THEN TV=TV+1:IF DA
=DE THEN CD=1
3609 IF DB=DC THEN TV=TV+1:IF DB
=DC THEN CE=1
3610 IF DB=DD THEN TV=TV+1:IF DB
=DD THEN CF=1
3611 IF DB=DE THEN TV=TV+1:IF DB
=DE THEN CG=1
3612 IF DC=DD THEN TV=TV+1:IF DC
=DD THEN CH=1
3613 IF DC=DE THEN TV=TV+1:IF DC
=DE THEN CI=1
3614 IF DD=DE THEN TV=TV+1:IF DD
=DE THEN CJ=1
3615 IF YA=1 GOTO 3616 ELSE GOTO
 3620
3616 RETURN
3620 IF U=222 GOTO 3630:ELSE IF
U=254 GOTO 3640:ELSE IF U=286 GO
TO 3650:ELSE IF U=318 GOTO 3660:
ELSE IF U=350 GOTO 3720:ELSE IF
U=382 GOTO 3740:ELSE IF U=414 GO
TO 3760
3630 IF TV<3 THEN TZ=0 ELSE TZ=T
Z:GOTO 3380
3640 IF TV<6 THEN TZ=0 ELSE TZ=T
Z:GOTO 3380
3645 GOTO 3380
3650 IF TV=4 THEN TZ=25 ELSE TZ=
0
3655 GOTO 3380
3660 IF TV=0 GOTO 3674 ELSE 3661
3661 IF CA=1 THEN DB=0
3662 IF CB=1 THEN DC=0
3663 IF CC=1 THEN DD=0
3664 IF CD=1 THEN DE=0
3665 IF CE=1 THEN DC=0
3666 IF CF=1 THEN DD=0
3667 IF CG=1 THEN DE=0
3668 IF CH=1 THEN DD=0
3669 IF CI=1 THEN DE=0
3670 IF CJ=1 THEN DE=0
3671 TZ=DA+DB+DC+DD+DE:IF TZ=14
GOTO 3690 ELSE 3680
3674 IF TZ=17 OR TZ=18 THEN TZ=0
 ELSE TZ=30
3675 GOTO 3705
3680 IF TZ=10 OR TZ=18 THEN TZ=3
0 ELSE TZ=0
3681 GOTO 3705
3690 IF DA=6 OR DB=6 OR DC=6 OR
DD=6 OR DE=6 THEN TZ=0 ELSE TZ=3
0
3705 IF TV>1 THEN TZ=0
3710 GOTO 3380
3720 EA=0:EB=0
3721 IF TV=0 THEN EA=1
3722 IF TZ=15 THEN EB=1
3725 IF TZ=20 THEN EB=1
3727 IF EA+EB=2 THEN TZ=40 ELSE
TZ=0
3740 GOTO 3380
3760 IF TV=10 THEN TZ=50 ELSE TZ
=0
3770 GOTO 3380
3800 M$=CHR$(252)
3803 PRINT@160,STRING$(32,M$);
3805 PRINT@129,"   ENTER PLAYER
NUMBER 1-4";:INPUT"";A$
3810 IF A$="" THEN GOTO 3805
3815 IF VAL(A$)<1 OR VAL(A$)>4 G
OTO 3805 ELSE 3817
3817 IF CL=0 THEN CL=100
3820 IF A$="1" THEN GOTO  3840:E
LSE IF A$="2" THEN GOTO  3850 :E
LSE IF A$="3" THEN GOTO  3860:EL
SE IF A$="4" THEN GOTO  3870
3830 IF VV=1 GOTO 3805 ELSE 70
3840 IF KX=0 THEN KA=XX:ELSE IF
KX=1 THEN KB=XX:ELSE IF KX=2 THE
N KC=XX
3845 KD=CL-100+KD:KX=KX+1:IF KX>
=4 GOTO 3880 ELSE GOTO 3890
3850 IF MX=0 THEN MA=XX:ELSE IF
MX=1 THEN MB=XX:ELSE IF MX=2 THE
N MC=XX
3855 MD=CL-100+MD:MX=MX+1:IF MX>
=4 GOTO 3880 ELSE GOTO 3890
3860 IFNX=0 THEN NA=XX:ELSE IF N
X=1 THEN NB=XX:  ELSE IF NX=2 THE
N NC=XX
3865 ND=CL-100+ND:NX=NX+1:IF NX>
=4 GOTO 3880 ELSE 3890
3870 IF RX=0 THEN RA=XX:ELSE IF
RX=1 THEN RB=XX:ELSE IF RX=2 THE
N RC=XX
3875 RD=CL-100+RD:RX=RX+1:IF RX>
=4 GOTO 3880 ELSE 3890
3880 PRINT@132,"       WRONG NUMB
ER":SOUND 10,10:FOR T=1TO500:NEX
T T:VV=1:GOTO3805
3890 ZA=KA+KB+KC+KD:ZB=MA+MB+MC+
MD:ZC=NA+NB+NC+ND:ZD=RA+RB+RC+RD
3900 PRINT@0,"      PLAYERS TOTAL
 SCORE"
3910 M$=CHR$(175)
3920 PRINT@32,STRING$(32,M$)
3930 PRINT@67,"PLAYER 1":PRINT@7
9,M$:PRINT@84,"PLAYER 2"
3940 PRINT@96," GAME 1   "KA:PRI
NT@111,M$" GAME 1    "MA
3950 PRINT@129,"GAME 2   "KB:PRI
NT@143,M$" GAME 2    "MB
3960 PRINT@160," GAME 3   "KC:PR
INT@175,M$" GAME 3    "MC
3970 PRINT@192," BONUS YTZ"KD:PR
INT@207,M$" BONUS YTZ "MD
3980 PRINT@224," TOTAL    "ZA:PR
INT@239,M$" TOTAL     "ZB
3990 PRINT@256,STRING$(32,M$)
4000 PRINT@291,"PLAYER 3":PRINT@
303,M$"    PLAYER 4"
4010 PRINT@320," GAME 1   "NA:PR
INT@335,M$" GAME 1    "RA
4020 PRINT@352," GAME 2   "NB:PR
INT@367,M$" GAME 2    "RB
4030 PRINT@385,"GAME 3   "NC:PRI
NT@399,M$" GAME 3    "RC
4040 PRINT@416," BONUS YTZ"ND:PR
INT@431,M$" BONUS YTZ "RD
4050 PRINT@449,"TOTAL    "ZC:PRI
NT@463,M$" TOTAL     "ZD;
4055 PRINT@495,M$;
4060 FOR T=1TO1000:NEXT T
4065 ED=ED+1
4066 IF ED=12 GOTO 4067 ELSE 407
0
4067 PRINT@0,"END OF GAME PRESS
<ENTER>";:INPUT"";A$
4068 CLS:END
4070 PRINT@0,"DO YOU WANT ANOTHE
R GAME Y/N";:INPUT"";A$
4075 IF A$<>"N" GOTO 4080 ELSE 4
068
4080 VV=0:GOTO70
4090 REM YAHTZEE BY FRANK WOODWA
RD BRISBANE 07-2007139
```

■

SCREEN DRIVER

By Russel Lucas
UTILITY
COCO 3

I'VE FINALLY DECIDED to send you one of the many programs that I have written over the last few months for the CoCo 3.

This program is a simple screen driver utility that allows you to display up to twelve programs on the screen at once.

An inverse video function is used to show the present cursor position and pressing <enter> will load the selected program.

The <esc> key can also be used at any point to return to Basic.

There are two data lines that must be entered by the user before running the program.

These are lines 430 and 440. Line 430 will contain the data that the user wishes to have displayed on the screen and line 440 is the actual file name of the program.

This is so you can have the actual name of a program displayed on the screen, even though the name may be abbreviated on the disk.

If you have less than twelve files that you wish to load then you must fill the remaining files with the minus sign (-) to show that space is not a program, eg:

```
430 DATA 1,2,3,4,5,6,7,8,-,-,-,-
```

The last four minus signs show that there is no program present for that space and if one is chosen then nothing will happen.

I originally wrote this program to support machine language programs only but it can easily be changed for Basic programs by changing the 'LOADM B$(T)' in line 210 to read 'RUN B$(T)'.

This utility is good to place on all your disks with the relevant programs displayed on the menu.

It is especially good for those who are unfamiliar with the computer and can easily type RUN "STARTUP" and have an easy to use screen driver.

The Listing:

```
0 GOTO100
3 SAVE"322:3":END'8
100 '*************************
110 '*                       *
120 '*    SCREEN DRIVER      *
130 '*                       *
140 '*   BY RUSSELL LUCAS    *
150 '*                       *
160 '*************************
170 '
180 GOTO 220
190 POKE 65496,0:SAVE"STARTUP/BA
S":DIR:STOP
200 '
210 IF A$(T)="-" THEN GOSUB 320:
GOTO 330  ELSE POKE65496,0:LOCAT
E X,Y:ATTR 0,0,B:PRINT A$(T);:LO
CATE 79,23:PRINT"";:LOADM B$(T):
WIDTH 32:POKE65344,0:EXEC
220 PCLEAR1:WIDTH 80:CLS5:PALETT
E 1,0:PALETTE 0,63:POKE 65497,0:
ON BRK GOTO 450
230 DIM A$(12),B$(12)
240 LOCATE 31,0:PRINT" SCREEN DR
IVER";:LOCATE 33,1:PRINT"MAIN IN
DEX";:LOCATE 22,23:PRINT"Select
Option, Press <ENTER> to Load";
250 FOR T=1 TO 12:READ A$(T):NEX
T:FOR T=1 TO 12:READ B$(T):NEXT
260 X=13:Y=8:T=1:C=3
270 IF X=30 THEN X=10
280 IF T=13 THEN 310 ELSE LOCATE
 X,Y:PRINT A$(T);
290 IF T=C THEN X=13:Y=Y+2:T=T+1
:C=C+3:GOTO 280 ELSE 300
300 X=X+20:T=T+1:GOTO 280
310 X=13:Y=8:T=1:C=1:A=13:B=8:LO
CATE 13,8:ATTR 3,1:PRINT A$(T);:
ATTR 0,0:LOCATE 79,22:GOTO 330
320 LOCATE X,Y:ATTR 3,1:PRINT A$
(T);:ATTR 0,0:LOCATE A,B:PRINT A
$(C);:LOCATE 79,23:PRINT"";:RETU
RN
330 A$=INKEY$:IF A$="" THEN 330
340 IF A$=CHR$(94) THEN 390
350 IF A$=CHR$(10) THEN 400
360 IF A$=CHR$(9) THEN 410
370 IF A$=CHR$(8) THEN 420
380 IF A$=CHR$(13) THEN 210
390 IF Y=8 THEN 330 ELSE A=X:B=Y
:C=T:Y=Y-2:T=T-3:GOSUB 320:GOTO
330
400 IF Y=14 THEN 330 ELSE A=X:B=
Y:C=T:Y=Y+2:T=T+3:GOSUB 320:GOTO
 330
410 IF X=53 THEN 330 ELSE A=X:B=
Y:C=T:X=X+20:T=T+1:GOSUB 320:GOT
O 330
420 IF X=13 THEN 330 ELSE A=X:B=
Y:C=T:X=X-20:T=T-1:GOSUB 320:GOT
O 330
430 ' PLACE DISPLAY DATA HERE
440 ' PLACE LOAD    DATA HERE
450 POKE 113,0:EXEC40999
```

■

COLOUR SCREEN DUMPS

By George McLintock UTILITY 32K ECB COCO + CGP-220

WHEN TANDY CLOSED the Computer Centre here in Canberra, they sold off a number of items which have been discontinued for some time. One such item was a number of ink jet printers, sold at a price that I found irresistible when I brought one.

The CGP-220 Ink Jet printer was never a big seller for Tandy. It was expensive (Over $1000 some years ago), and its print quality as a printer was not very good. However, as a colour printer for screen dumps it is superb.

The printer provides four basic ink colours, black, red, green and blue. A further three colours can be obtained by a single overprinting of the red, green and blue inks to get yellow (red & green), magenta (red & blue) and violet (green & blue). The eighth colour available is no colour (white or paper colour). The printer does not allow overprinting with the black ink.

While this is adequate for the 4 colours of Pmode 3, it is only half the number of colours available with Hscreen 2 of the CoCo 3.

The CoCo screen dump provided by Tandy for the printer was also quite limited. You could select one of three colours for the background, and this fixed the other three colours for the dump.

Eg, if you selected green for the background, then the other three colours were yellow, blue and red. This setup could not be altered.

After I got the printer, I set out to develop a more flexible screen dump that would provide more colours and allow any pixil colour to be set to any printer colour. Along the way I've also provided for different scaling factors for size of dump, and for a different coloured border and margins to be added to each screen dump. The dump will operate on any of the Pmode's or Hscreen's in the CoCo.

Although the programs themselves may be of limited interest (I don't think there are many CGP-220's around - and although six were sold in Canberra recently, at least two of these are not on CoCo's),

The procedures for generating more colours from the basic set may be of more general interest.

Also, while I have not checked the specific details of how other colour printers generate their colours as they move across the page, it should be fairly easy to convert these programs to suit any other printer that follows the same general procedure for printing colours.

Producing Colours with the CGP-220.

The CGP-220 has two basic modes of operation. In its normal mode, it operates as a standard Tandy printer with a colour option. Any of the 8 basic colours can be selected for normal printing and normal graphics, ie graphics as seven bits per row of print, with the high order bit always one.

In this mode, a normal Tandy printer screen dump will work without modification except for changing the control codes to start graphic printing.

If the colour is selected before the dump starts, you can get a single coloured dump by using any of these existing programs.

The single colour replaces the normal black for these dumps.

There are some other interesting options for printing in normal mode which are not standard for Tandy printers but these are not covered here.

The other mode of operation is colour scan mode, which effectively prints a single row of dots across the page, with the colour of each dot specified independently.

The colour of each dot is specified by a combination of three bits. If all three bits are off, the printer prints a black dot. If all three bits are on, then it prints no colour at all.

The first bit controls the red colour, the second bit the green colour, and the third bit the blue colour. If these bits are on then it prints that colour in the dot position.

Hence if the first bit only is on, then it prints a red dot, and if both the first and second bits are on then it prints both red and green to give a yellow in that dot position. As noted, if all bits are on then it prints no colour.

This setup provides for the 8 basic colours that can be produced by the printer.

The number of dots to be printed in colour scan mode can be varied. The printer allows for a maximum of 640 dots across the page, (which produces the 91 characters per line of normal text that can be printed).

For these screen dumps, I've set it up to print all 640 dots in each line, and set dots to white which are not required for the dump.

In this situation, each line of dots to be printed requires 240 bytes to be sent to the printer (plus 3 bytes for control codes). The first 80 bytes contain the bits that control the red colour for each of the 640 dot positions across the page. Each bit in these 80 bytes corresponds to the dot positions for the printer starting from the left. The next 80 bytes control the green colour, and the last 80 bytes are for the blue colour.

The dump operates by starting with a string of 240 bytes with all bits zero. A machine language routine then sets the bits on as required to dump that

line of dots, and the string is then printed with a PRINT#-2 command in Basic.

Getting more Colours

The 8 basic colours are produced by printing each dot in the colour desired, so that it fills an area of the paper in that colour to correspond with an area of colour on the screen.

Whilst this is obvious, it forms the basis of describing how other colours are produced.

For example, if the area of the screen for (1,1)-(10,10) is red, and you want this area to be red on the paper, then you print all corresponding dots on the paper in red. ie

```
R R R R R R R R R R
R R R R R R R R R R
R R R R R R R R R R
```

... etc for 10 rows

I have developed four variations to this normal pattern which are outlined below. The detailed logic is covered later.

Two Coloured Patterns

The first variation to this is simply alternating colours, which I have called two colour patterns. For example a lighter red can be produced by alternating red and white dots as follows:

```
R W R W R W R W R W
W R W R W R W R W R
R W R W R W R W R W
```

... etc for ten rows

The general form of this pattern allows you to mix any two colors, and the 8 basic colors allow 28 different patterns (7+6+5+4+3+2+1)

eg red and black gives a darker red, while red and green give a sort of variation of yellow to a brown.

Multiple Colour Patterns

The next variation is a simple moving colour pattern which I have called multiple colours. eg a 4 color pattern of red, green, blue and yellow would be ...

```
R G B Y R G B Y R G
G B Y R G B Y R G B
B Y R G B Y R G B Y
Y R G B Y R G B Y R
```

For this one I have provided two options for how it repeats at the end of the basic pattern (ie after 4 rows). The first option continues to repeat the same lines from the start, while the second option counts back one colour at a time

eg for option 1 the next 4 lines would be ...

```
R G B Y R G B Y R G
G B Y R G B Y R G B
B Y R G B Y R G B Y
Y R G B Y R G B Y R
```

While for option 2 they would be ...

```
B Y R G B Y R G B Y
G B Y R G B Y R G B
R G B Y R G B Y R G
G B Y R G B Y R G B
```

The general form of this pattern allows for a very large number of different colour patterns to be produced. The colour sequence can be of any length, and it can be set to repeat after any (different) number of rows. The fixed bit of logic for this one is that each line starts with a different dot colour to the one before it.

"This program will print pictures in eight colours"

Colour Blocks

The third variation removes the restriction that the colour changes for each line printed. I call it colour blocks.

There are effectively two variations of this pattern depending on the value of the parameters set.

The first variation is effectively an extension of the two colour pattern to blocks of colours that can extend to multiple colours.

For example, a two colour block of three colours, red, green and blue could take the following form ...

```
R R G G B B R R G G
R R G G B B R R G G
G G B B R R G G B B
G G B B R R G G B B
B B R R G G B B R R
B B R R G G B B R R
```

Within this variation there is another option to have the rows repeat in the same fashion, or to count back by colour blocks. Follow this along similar lines as this option obtains multiple colours.

Again there are a very large number of patterns that can be built up using this procedure. It has the same flexibility for varying the different parameters as provided for multiple colours.

The second variation allows for stripes down the page. In effect the colours don't move at all. (The increment is zero). Using the example above for stripes would be ...

```
R R G G B B R R G G
R R G G B B R R G G
R R G G B B R R G G
```

... without any change

Row Colours

The final variation included is one which allows rows of colours across the page. eg 3 red, 3 green, 3 blue etc. Again in any combination to provide a large number of choices.

Parameters for Colour Paterns

Each colour pattern is set up as a series of parameters which are accessed by a ML routine to determine the color of the next dot to be printed.

The parameters for each of the variations are specified below, together with a brief outline of the logic used for the print operation.

This should help clarify the various options that can be obtained.

Parameters common to all options are ...

.This screen pixil

The printer colour patterns are set up independently of the screen dump to be performed. This parameter links the printer colour pattern to a screen pixil colour for the dump, eg if the parameter is 2, then this colour pattern will be used for the screen pixil colour that equals 2 when it is extracted from screen memory.

As borders and margins for the dump don't have a corresponding screen pixil number, I use numbers greater then 64 to

represent these areas of the dump, where the same colour is not used as a screen pixil.

This also allows the same printer colour to be used with different screen pixils, eg if pixil 1 and pixil 2 are both set to the same printer colour (Multiple copies of the same printer colour are allowed), then they will both appear as the same colour in the dump.

A similar procedure can be used to eliminate colours from the screen. By setting them to the same printer colour as the background, they will not show in the dump.

.This colour

Represents the position number of the next colour in the pattern to be printed. It is a signed number from zero to the number of colours in the pattern, and is updated after each dot is printed.

.Start colour

Represents the position number of the colour to be used for the first dot in the next line printed. The 'this colour' parameter is set to the start colour value at the beginning of each line printed, and the start colour value is then updated for the next line.

The number of parameters for each colour pattern within a pattern type, has to be the same so that they form a table which can be searched by the program. The two colour pattern has only two colours and these are stored as part of the parameter list.

However, the other patterns allow for a variable number of colours in each pattern, and the actual colours to be used are stored in a different part of the memory. This also allows the same series of colours to be used by more than one colour pattern.

Other parameters common to these three options are:

.Start address of colours

Contains the memory address where the colours for this pattern start.

The parameters for start colour, this colour etc are used as displacements from this address to find the actual printer colour to be used for the dot to be printed.

As coded, the displacement is a signed value, so that if the number of colours exceed 127, you have to arrange the colours in memory to suit.

.Number of colors in the pattern

This is used in association with the start colour to determine when to reset the start colour to the begining of the colour sequence.

Each time the start colour is altered, it is compared with the number of colours in the pattern, and is reset to zero when it reaches this value. Specific parameters are ...

Two Colour Patterns

0=this screen pixil
1=start colour
2=this colour
3=first colour in pattern
4=second colour in pattern

The this colour value alternates between zero and one after each dot is printed across the line.

The start colour alternates between zero and one after each line is printed.

Multi-Coloured Patterns

0=this screen pixil
1=start colour
2=roll start
3=switch for up/down
4=this colour
5=number of colours
6&7=start address of colours

Each time a dot is printed across the page, this colour is incremented by one and compared with the number of colours in the pattern. When they are equal, this colour is reset to zero.

For each line printed

- If the switch for up/down is zero, then the start colour is increased by one and compared with the roll start value. When they are equal, the start colour is reset to zero
- If the switch is positive, then the start colour is increased by one and compared with the roll start value. When they are equal, the switch is set to a negative value
- If the switch is negative, then the start colour is reduced by one and compared with zero. When it becomes zero, the switch is set to positive again.

Colour Blocks

0=this screen pixil
1=start colour
2=roll row
3=switch for up/down
4=this colour
5=number of colours
6&7=start address of colours
8=this increment
9=roll increment
10=colour increment for each roll

Each time a dot is printed across the page, this colour is incremented by one and compared with the number of colours in the pattern. When they are equal, this colour is reset to zero.

For each line printed

- if the up/down switch is zero
- this increment is increased by one and compared with the roll increment value
  : if less than, no other action occurs
  - when it equals the roll increment value
  : this increment is reset to zero
  : the start colour is increased by the colour increment for each roll
  : and the start colour is compared with the roll row value and if it is less than, then no further action occurs
  : when the start colour reaches the roll row value, the start colour is reset to zero
  : If the up/down switch is non-zero then positive and negative values are used in a similar way as for multiple colours
  : ie the start colour is increased or decreased by the colour increment for each roll
  : and compared with the roll row value or zero depending on the switch setting
  : when it reaches its limit, the switch is altered to count in the opposite direction for the next change.

Row Colours

0=this screen pixil
1=start colour

Tandy ELECTRONICS

Update Your Computer System and Save

10 Megabyte Disk Cartridge

CLEARANCE

Save $3000

$1999

Reg 4999.00

- ***Ideal to Use for Primary Disk Storage or Backup***

Update your computer system and save an incredible $3000! Enjoy portability, speed, reliability, expandability and security without compromise! Ideal for use as a primary data storage device instead of a hard disk drive. Run an unlimited amount of data using the removable 10-Megabyte Cartridge system without adding drives. Includes cartridge. (Sorry no rainchecks). 26-1245

Tandy V21/23 Modem

Save $100

299.95

Reg 399.95

Services through Telecom's Viatel, 300 bits/sec or 1200 bits/sec receive. 26-9404

Quality Internal Hard Disk

Save $2700

Includes Free Installation

$1499

Reg 4199.00

Gain access of up to 15 million characters of data! Upgrades models 12 and 16B. 26-4154

External Intelligent Modem

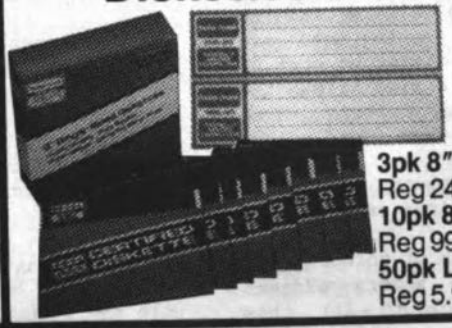
Save $200

$549

Reg 749.95

Talk to other computers, bulletin boards, information services and data bases. Use with our specialised software for auto-dial/answer functions! 26-9419

Diskettes and Labels



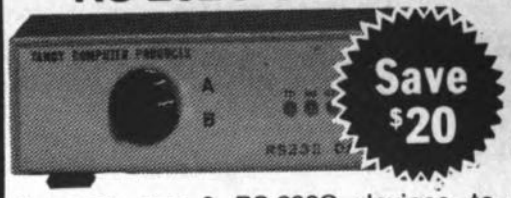
SAVE up to $20

3pk 8" Diskettes. 26-4904
Reg 24.95 . . **Sale! 14.95**

10pk 8" Diskettes. 26-4906
Reg 99.95 . . **Sale! 79.95**

50pk Labels. 26-4908
Reg 5.95 . . . **Sale! 2.95**

RS-232C Selector Switch



Save $20

99.95

Reg 119.95

Connect any 2 RS-232C devices to your Tandy Computer. With 2 switched ports for modems. Configurated port for communications between 2 computers (cables not included). 26-9420

Convenient VM-1 Pedestal

Save $80

99.95

Reg 179.95

Monitor not included

Compact and stylish with grooves for a strong grip. Elevates your computer, swivels. 26-5115

External Disk Drive

$299

Reg **499.00**

SAVE $200

CLEARANCE

Increase the disk storage of your Model III Desktop Computer. High-capacity 184K external double-density 40-track 13.3cm mini-disk drives. Internal Drive #1 required. 15.8 x 8.8 x 36.8cm. (Sorry no rain-checks). 26-1161

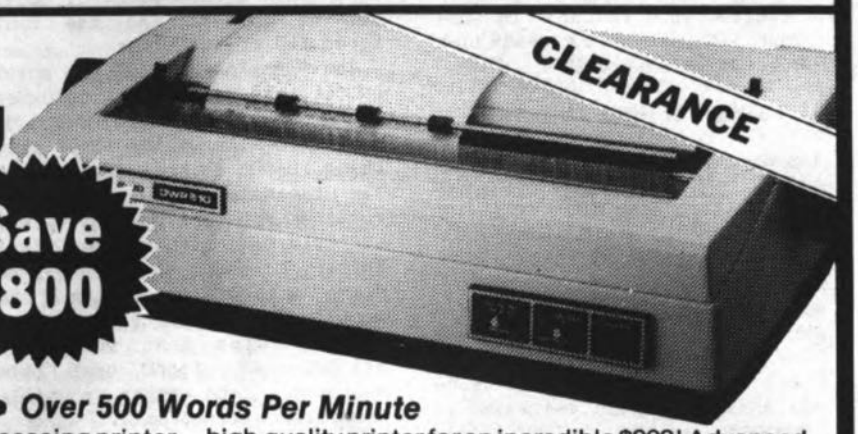
The Daisy Wheel Word Processing Printer

$999

Reg **1799.00**

Save $800

CLEARANCE



- ***With Clean Sharp Characters For a Professional Look!***
- ***Over 500 Words Per Minute***

DWP-510. It's the professional word processing printer for high performance and an excellent $800 saving! Our DWP-510 word processing printer is designed for speed and accuracy printing over 500 words per minute, so that you'll never have to wait for the printer again. There's no need to put up with second quality printouts when you can get all the advantages of a high-quality printer for an incredible $999! Advanced features include an interchangeable 124-character print wheel, proportional spacing (switch selectable) and so much more! Prints up to 5 copies with a courier 10 print wheel and carbon ribbon. Use our professional word processing printer and be proud of the excellent results! (Sorry no rainchecks). 26-1270

Clearance on Tandy 1000 & CoCo 2 Software

Friday. Can be used with Lotus 1-2-3 (for T-1000). 25-1149 Was 749.95 . . . . . **Now! 149.95**

Microsoft Multiplan. An easy to use spreadsheet analysis program! Ideal for profit and lose projections, budgeting and more! 25-1152 Reg 399.95 . **Sale! 179.95**

Sale! T-1000 Scripsit Word Processing Software. Advanced features. Merge files, boilerplating, multiple fonts. 25-1155 Reg 699.95 . . . . . **Sale! 369.95**

Micro-Illustrator. Create sophisticated drawings with a variety of brush strokes. Requires 64K CoCo 2, disk drive, joystick of mouse. 26-3278 Reg 39.95 . . . **Sale! 14.95**

Shamus. A space-odyssey game of excitement and adventure! Work your way through a maze of passages and rooms. Requires a disk drive and joysticks (for CoCo 2). 26-3289 Reg 39.95 . . . . . **Sale! 29.95**

TRS-80 Color LOGO Program Pak. Teach children to program through the manipulation of a "turtle" graphic on the screen. (For CoCo 2). 26-2722 Reg 79.95 . **Sale! 19.95**

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

or Order On VIATEL *642614#

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

2=roll row
3=row count
4=this colour
5=number of colours
6&7=start address of colours

These parameters are not altered after each dot is printed. The same value of the colour is used across the page.

For each line printed, the row count is increased by one. When it reaches the roll row value, the start colour is increased by one, and the row count reset to zero.

As with all patterns, when the start colour reaches the number of colours in the pattern, it is reset to zero.

A switch to count up and down the colours is not included with this pattern because it is easy to include this feature in the colour patterns without requiring extra code.

Normal Printer Colours

There is another table in the system for normal printer colours.
It consists of ...

0=this screen pixil colour
1=printer colour

Features Common to all Patterns

Printer colours are represented by the values 1 to 8, where ...

1=red
2=green
3=blue
4=yellow(1&2)
5=magenta(1&3)
6=violet (2&3)
7=black
8=white

The values in the colour patterns themselves must be standard printer colours (1 to 8). If a value is outside this range, it will be printed as white (colour 8 - no colours)

I did consider allowing colour patterns to be set up in terms of other patterns, but that becomes a bit complicated and seems a little excessive at this stage.

Program Operations

The screen dump itself is a Basic program which makes extensive use of machine language routines for most of the work. The high level control logic and setup of parameters is coded in Basic, with the ML routines used for the low level detailed data manipulation.

The dump is performed by a sequence of two quite separate operations.

The first operation extracts the pixils from the graphic screen memory and puts them in the pixil buffer, where each pixil occupies a separate byte of memory.

The second operation then sets up the print buffer as a completely separate exercise. The print buffer is 240 bytes long, and forms part of the string (A$) which is printed by Basic. This string (A$) is 243 bytes long, where the first three bytes are control codes for the printer, and the rest forms the print buffer.

The setting up of the print buffer, and printing of coloured pictures can in fact be performed without the first part being used at all. So long as the parameters for the ML routine are set correctly, and suitable data put in the pixil buffer, coloured pictures can be drawn without accessing the graphic screen memory.

The program that prints out all colours defined uses this approach, and other possible uses are outlined later.

Setting up the pixil buffer

This folows a conventional procedure to extract the bits for each pixil in the graphic screen and putting them into separate bytes in the pixil buffer. The number of bits per pixil and bytes across the screen for the different graphic screens are set up as parameters in the direct page before the ML routines are entered.

Separate entry points are provided to extract the pixils from either across the screen or down the screen, depending on the parameter vaues set.

Setting up the print buffer

The print buffer is set for 640 dots across the page for all dumps. The ML routines move across the line, setting each dot colour as it goes. The left margin and border are set first, then the dots are set for each pixill in the pixil buffer, and finally the right border and margin.

By changing the size of the left and right margin, it is possible to locate the dump anywhere across the page

For each line set in the print buffer, the program updates the parameters for all colour definitions as described above. Similarly, as each dot is set the parameters are also updated as described for each dot. This is necessary to ensure that the colour patterns print consistantly across and down the page.

Each dot to be printed has a value associated with it, which is either the value of the pixil as extracted from the graphic screen, or a value to represent one of the borders or margins. Once this value is selected, the routine then searches the table of colour definitions to find the matching printer colour that corresponds with that value, (this value is the first parameter for each colour definition).

When the matching printer colour is found, the routine then picks up the value of the corresponding basic printer colour to be printed, (also included in the list of parameters for each colour definition), and sets the bits in the print buffer to print that colour in the dot position.

In all cases the updating of the parameter list for each colour definition includes the value of the next basic printer colour to be used, (it is called 'this colour' in the parameter list, and is parameter 4 in all colour colour definitions except the two colour one).

This approach provides a convenient way of breaking down the logic for the whole operation into reasonably sized components for coding and testing.

It also means that a large part of the overall program logic is completely independent of the specific colour printer being used and how it actually controls its printing of colours. The only change required to suit any other printer that also prints a single row of dots across the page, is to the small routine which actually sets up the bit pattern in the print buffer to generate the colour required in each dot position.

If no printer colour definition is found to match the

value being searched for, the printer colour for that dot is set to white (or no colour).

Memory Usage

While the ML code itself is relocatable, the way in which the colour patterns are defined is not. Although the colour patterns could be made relocatable, it hasn't seemed to be worth the effort at this stage. Hence the ML routines must occupy a fixed area of memory and cannot be incorporated at the end of a basic program

As now set up, the ML routines and colour definitions occupy protected memory above address 26000. The ML routines themselves occupy less than 1000 bytes and the remainder is used for colour definitions. While this is reasonable for a 32K machine, it could be compressed to fit in a 16K machine if nesessary. The CoCo3 routines would not be required, and memory usage could be reduced considerably by reducing the space for colour definitions.

The two main dump routines are stored from 26000 upwards, with the colour definitions following. Space is available for 28 two coloured patterns, 25 row colours and 100 each for multi colours and colour blocks. There is still approx 1K above this for other uses.

In actual fact this is not a very good way of organising things, but it works, and at this stage I don't feel any real need to change it.

The ML routines use a large number of parameters and tempory storage for variables. To make the code relocatable, I've used the direct page register to point to this area, and located the direct page for this purpose with respect to the Pmode graphic screen pages.

Reference addresses are stored by Basic at Hex BA for the start of graphic screen, and Hex B7 for end of graphic screen.

For the Pmode 3 dump, the program sets Pclear 5, and uses the fifth graphic page as the direct page for data storage.

The method used to access the CoCo 3 graphic screen requires all program code and data used to be located below address Hex 2000. This is achieved by setting PMODE 0,1, which then locates the direct page for data storage in the second graphic page. (The start of the direct page is taken from location Hex B7).

The program code used to extract the pixils from the CoCo 3 screen is then moved to the first graphic page (Obtained from Hex BA) and executed from there.

As both the pixil and print buffers are set to follow the direct page area of memory, this effectively puts both these buffers below Hex 2000 for the CoCo 3 dumps.

Improved Memory Storage

Most of the problems associated with memory usage are associated with the way I've defined the different colour patterns, with these programs being developed in a sequential fashion.

With the knowledge gained from doing it this way, I feel it would be better to use the same number of parameters to define all colour patterns. It could be done with 12 parameters and the same 11 as for colour blocks plus one to show the type of pattern to be used

While this would waste some memory for the definition of all patterns, it would allow the definitions to be grouped into a single table, with a single area of memory devoted to it, and another single area of memory to be set aside for colour specifications.

The way it is now set up is that sufficient memory is provided in a single block for a maximum number of each colour patterns, repeated four times.

Also with a single table, it would be more reasonable to code the definition of patterns to be relocatable.

But at this stage I don't intend to recode it that way, perhaps later on.

Size and Rotaion of Dump

The 640 dots across the page allow for the doubling of the size of the dump for the Hscreens 1 & 2, and for two and a half size for the Pmode 3 screen (with normal rotation). If the dump is rotated 90 degrees then all screens can be dumped at three times normal size. These dumps allow for the three times size.

The scaling up of the dump is applied at the time that the pixils are extracted from the graphic screen. For example to double the size of Hscreen 2, the pixil buffer is set to 640 bytes, and each pixil extracted from the graphic screen is stored in the pixil buffer twice.

The actual printing of the dump is done with a FOR .. NEXT loop in Basic. At double size, for each of the 192 lines down the screen,

- a line of pixils is extracted from the graphic screen and put in the pixil buffer, with each pixil duplicated in the pixil buffer,
- to get a doubling of size down the page, the routine to set up the print buffer is then executed twice, using the same set of pixils each time.

In this situation you can't simply print the same print buffer twice, because the colour patterns will not be updated if you do. The execution of the routine to set up the print buffer is required to update the parameters for each definition.

For the Pmode 3 screen, each screen pixil is in fact two dots wide for a normal size dump, and only one dot deep. It is this aspect which allows the screen to be dumped at two and a half size. For this size, each pixil is extracted five times into the pixil buffer to get 640 dots wide.

To get the two and a half size down the page, the dump alternates between 2 and 3 repeats for each line of pixils extracted from the screen.

OTHER SIZE OPTIONS

Larger sized dumps can be obtained if parts of the graphic screen are excluded, eg a quarter of the Hscreen 2 screen could be dumped at four times size with normal rotation.

For this dump I've provided for these options by allowing margins (left, right, top, bottom) to be set for the graphic screen. The dump skips the margins and only operates on the area of the screen inside the margins. This is achieved by only taking the pixils from inside the margins to the pixil buffer.

Likewise an option is included to do any sized dump rotated 90 degrees, or with normal rotation.

Borders and Margins

All borders and margins for the dump consist of four parts, left, right, top and bottom. Normally the margins would be white with a singe coloured border around the whole picture.

However to allow some experimenting with unusual dumps, I've allowed each separate part of both borders and margins to be specified as different colours and to be of different sizes.

As coded, it is quite easy to allow for these options, so I did.

Mirror Image Effects

A mirror image effect is fairly common with screen dumps. It occurs when you transfer the x,y co-ordinates of a point for one device (computer screen) to the same x,y values for another device (printer or plotter) which measures these values from a different origin, eg the origin for the CoCo screen (point 0,0) is the top left corner of the screen.

If you draw a picture on the CoCo screen and then draw the same picture using the same x,y values on a plotter which has its origin (point 0,0) in the top right corner of the page, then the picture on the plotter will be a mirror image of the picture on the screen.

If you take the picture from the plotter and hold it up in front of a mirror, you will see the picture in the mirror the same as you see it on the screen.

While a printer doesn't have a specific origin for a picture in the same way as a plotter does, the sequence in which the dots are printed effectively operates in the same way.

For these dumps, rotating the screen 90 degrees will produce a mirror image effect, if the co-ordinates for the points extracted from the screen are not modified.

There are various options available for correcting mirror image effects produced during screen dumps. The one used here is to adjust the pixils in the pixil buffer after they have been extracted from the screen, ie exchange the first and last, second and second last etc.

An incidental effect of doing it this way is to allow the deliberate creation of a mirror image effect for those dumps where it would not normally occur, ie to allow any dump to be a mirror image of whats on the screen. Hence, because it is easy to do, I've included it as an option for all dumps.

Set-up of Parameters

The routines used for this dump have been coded to handle all CoCo screens, but for them to work correctly, a number of parameters have to be set to appropriate values first. Most of the parameters apply to the ML routines and these are poked into the parameter table before these routines are executed

"The routines used for this dump have been coded to handle all CoCo screens."

I have also developed a small Basic program to calculate these parameters from a menu arrangement to enter the basic information. eg size, graphic screen, setting pixil to printer colours, etc.

However, to keep this part simple, I've restricted the automatic set up of parameters to Pmode 3, and Hscreens 1 and 2. The other screens, including the two colour ones can be dumped, but I have not included them in the set up program. To dump these, parameter values have to be poked into the ML routines separately.

Selection of Colours to be Used

A problem with this approach to colour dumps is to achieve some trade off between the number of different colours which can be obtained, and the size of the definition tables to be searched and updated during the dump. The greater the number of colour patterns to be updated during the dump, the longer it will take to complete the dump.

The approach adopted here is to use a separate block of memory to define the colours, and then to select the colours required for the dump and move the parameter list for those colours from the colour definition area into a separate 'execute time' table which is then updated during the dump.

To take the colour block patterns as an example. There is an area of memory set aside to allow up to 100 different colour block patterns to be defined, but it would take an excessive amount of time to update all of these for each dot printed during the dump.

To avoid this, there is a separate 'execute time' table set up to hold the colour patterns which are actually required for the dump. And it is this execute time table which is actually used and updated during the dump

This procedure also allows the same printer colour pattern to be associated with more than one pixil colour, eg both pixil 1 and pixil 2 can have the same printer colour. In this case, two copies of the printer colour parameters are brought across to the execute time table.

Part of the set up routine includes the selection of colour patterns in the colour definition area and moving those required to the execute time tables. The parameter list for each pattern can, in fact, be operated on at any memory location, which allows this move to be done.

The only part of the system which prevents the dump being fully relocatable at this stage is the parameter for the start address of the colours. To make the whole thing relocatable would only require the ML initialisation routine to loop through all the colour definitions tables and adjust these parameters to the actual memory locations used at the time.

An alternative setup arrangement could be to restrict the number of colours to a limited set in the execute time tables only, and leave them there all the time. This would also substantially reduce the memory required for colour definitions, but would make it more difficult to achieve the same printer colour being used for more then one pixil colour.

Another alternative, of course, would be to define all colours required at the start of each dump.

The setup actually used can

also be altered quite easily to vary the number of each pattern type that can be defined, ie to increase the number of colour blocks that can be defined.

part three (318b on disk9)

PARAMETERS REQUIRED FOR DUMP

A number of parameters are required to be set for the ML routines before they are executed. These parameters are contained in either the direct page, or as part of the ML code itself.

In the list that follows parameters in the ML code are shown as actual Hex addresses, while those in the direct page are shown as Q + 'value', where Q = PEEK(&HB7)*256 (Start of direct page)

Q+47=colour of left margin
Q+48=low order byte of number of dots in left margin
Q+49=high order byte of number of dots in left margin
Q+50=colour of right margin
Q+51=low order byte of number of dots in right margin
Q+52=high order byte of number of dots in right margin
Q+53=colour of left border
Q+54=number of dots in left border
Q+55=colour of right border
Q+56=number of dots in right border
Q+57=colour of top/bottom border/margin

The value poked in here is used to replace the values of pixils in the pixil buffer when doing the top and bottom borders and margins.

Q+63=number of times to repeat each pixil in the pixil buffer
Q+64=pixils per byte
Q+65=bits per pixil
Q+70=number of bytes across the graphic screen
Q+71=number of bytes in left margin of screen
Q+72=number of bytes to use for extracting pixils from the graphic screen and moving to pixil buffer
Q+73=number of bytes in right margin of screen

When extracting pixils down the screen, the values in Q+71 to Q+73 represents pixils down the screen, rather than bytes across the screen.

The address of the start of the next line of pixils to be extracted from the graphic screen is stored at Q+59 and Q+60. This is normally set up and maintained by the ML routines, but can be altered if you want to perform the same operation from another area of memory.

Hex 68F6 and 68F7 contain the number of pixils in the pixil buffer. This value must be set before the initialise routine is executed. It is used to calculate the end of the pixil buffer address, and is set to 256 in the ML code.

Entry Addresses

Hex 67C0 = reverses pixils in the pixil buffer to correct or generate a mirror image effect
Hex 67ED = extracts the next line of pixils from Pmode graphic screen
Hex 6829 = skips a line across the screen
Hex 6833 = skips a line down the screen
(Used for top and bottom screen margins)
Hex 6849 = extracts pixils down screen (Pmode)
Hex 68AE = initialises the routine for the Pmode screen
Hex 689B = initialises the routine for Hscreens

The initialise routine is called by A$=USR(A$) with DEFUSR0 equal to one of the values above.

The initialise routine also sets the VARPTR of A$ to point to the print buffer Hex 6590 = fills pixil buffer with border colour Hex 665E = execute address
to set up the print buffer

When extracting pixils from the Hscreen, the ML routines which access the graphic screen are moved to the Pmode graphic screen and executed from there. (This move is done by the ML initialisation routine). The entry points there are at an offset from Q1 where Q1 = PEEK(&HBA) * 256

Q1+0 = extracts pixils from across screen
Q1+180 = extracts pixils from down the screen

The routines to skip screen margins works the same for all screens. All it does is to change pointers.

The number of colour patterns for each execute time table is normally set in high memory before the routine is initialised. As part of the initialisation routine, the table of pointers to the execute time tables of colour definitions
is moved to the direct page and accessed there.

Hence these values exist in two places in memory.

Q+76 & Hex 68FE = number of two coloured patterns
Q+81 & Hex 6903 = number of multi coloured patterns
Q+86 & Hex 6908 = number of block patterns
Q+91 & Hex 690D = number of row patterns
Hex 66AD (only) = number of normal printer colours

Setting up Colour Patterns

A separate program, called SETUP, is used to actually set up the colour definitions in memory, and to print all colours that are defined.

It contains four components which are accessed through a main menu.

1. A routine to allow new colours to be defined in memory. Using data entered through the keyboard. This allows individual colours to be set up following a simple menu.
2. A series of routines to define a number of patterns using Basic programming code.

Defining colours through the menu becomes tedious after a while and it is quite easy to set them up with a few lines of Basic code.

This part of the program can be modified as required to set up any colours required. It also establishes the area of memory to be used for the definition of each pattern type, and hence the maximum number of each pattern that can be defined. If this is changed then the corresponding addresses in INKDUMP must also be changed to the same values.

3. Prints all colours defined in memory to a printer using character format. It prints out the reference code, parameter values, and colour definition values as they exist in memory.

4. Prints all colours defined to the GCP-220 in the way that they would appear in a screen dump.

The colours are printed in a box, 80 dots wide by 20 dots deep, with 8 boxes per line. The reference to the colours is printed below each box.

For this one, the program sets up its own pixil buffer of 640 pixils wide, and sets up pixil values of from 0 to 7 in this buffer. It then cycles through all the colours defined, bringing eight colours at a time into the execute tables.

Each colour in the execute table is then set to a pixil value of 0 to 7, and twenty lines are dumped to the printer, using the second part of the dump routine only.

Combining Pictures for a Single Dump

Normally screen dumps are done from a single screen picture to a single picture on the paper. However, there are times when you might want to combine a series of related screen pictures into a single large dump.

For example, I quite liked the series of Footy Badges by Joy Wallace in Softgold, March 87, and have used these to produce a single large dump of the 14 badges together, as three across the page for four rows, and the last two as two across the page below them.

This required a certain amount of additional programming using direct access files on disk, but the general set up and operating logic for these dump routines allows for this type of activity to be undertaken without too much effort.

The program for Footy Fever is set up to draw the 14 badges in sequence on the Hscreen 2 of the CoCo 3.

After each badge is drawn, the program waits for the break key before drawing the next one.

Some additional code was added to the program so that, instead of waiting for the break key, it executed a GOSUB 2000, where each screen was then extracted into a separate direct access file on disk.

The file for each screen has a record length of 160 bytes (number of bytes per row of pixils) and contains 192 records (number of rows down the screen).

"This program can handle dumps of multiple pictures."

The extraction of bytes from the graphic screen follows the same general procedure as used to extract the pixils into the pixil buffer. Only instead of extracting the pixils at this stage, the routine simply moves 160 bytes at a time from the graphic screen to the direct access buffer for the disk. (Starting from Hex 989).

After each record is extracted, it is PUT to the disk file.

I have also included a copy of the code used to do this. It it could also be useful for other purposes.

As well as extracting graphic screen memory into a direct access disk file, it also produces a table to show the colours used for each palette slot, and the number of pixils set to each palette.

This saves working your way through the program listing to find out which colours are actually used and not used.

This second part of the routine can also be used on any Hscreen 2 picture without actually saving the screen to disk.

The Pmode 3 (& 4) screen can be extracted and put into a direct access file without any ML routines at all. By changing the VARPTR of strings, I've also included a small Basic program to show how this can be done.

The next stage was to combine a record from each of these files into a single record for the dump.

When drawn, each badge occupies the centre of the screen with a reasonable margin. It is possible to fit three badges across 640 dots by taking 212 pixils (106 bytes) from the middle of each screen and combining them into a single row of pixils across the page, (with either a 2 dot margin on both sides, or making the middle one 216 pixils wide).

This can be done reasonably simply with a disk system by using the FIELD and LSET commands with four files open.

File #1 is opened first and is the result file, fielded as

```
FIELD #1, 106 AS AP$, 108 AS BP$, 106 AS CP$
```

File #2 is the first badge extracted and is fielded as

```
FIELD #2, 27 AS A1$, 106 AS A2$, 27 AS A3$
```

File #3 is the second badge extracted and is fielded as

```
FIELD #3, 26 AS B1$, 108 AS B2$, 26 AS B3$
```

File #4 is the third badge extracted and is fielded as

```
FIELD #4, 27 AS C1$, 106 AS C2$, 27 AS C3$
```

The first row of the result file is then produced by

```
GET #2,1: GET #3,1: GET #4,1
LSET AP$=A2$: LSET BP$=B2$:
LSET CP$=C2$
```

This produces a single row of 320 bytes, starting from Hex 989 and containing 640 pixils in the same format as if they were stored in a single row of graphic memory for Hscreen 2. (It also requires a FILES 4,820 to work.)

The extract routine is then set up to extract from a graphic screen with 320 bytes across each line, 2 pixils per byte and 4 bits per pixil.

These are all separate parameters poked into the direct page table, and the routine will operate quite normally with these parameter values. The

pixil buffer is set to 640 pixils wide to suit.

The extract routine has a parameter (at Q+59) in the direct page to show the start of the next line of pixils to be extracted. After each row is set up in buffer #1, Hex 989 is poked into this pointer and the extract routine executed as if for the pmode screen. This puts the pixil values into the pixil buffer , and the second stage of the dump is executed normally.

This same general procedure can also be used to scale up each badge to a larger size, by using the data from a single disk file as the result file.

For example, you can get a badge three times normal size by setting the extract routine to extract from a screen 160 bytes wide, 2 pixils per byte, 4 bits per pixil, a left screen margin of 27 bytes, extract 106 bytes, and scale up by three. For this one a 4 pixil margin must be provided, and the right screen margin set to 27 bytes.

There are other ways of achieving the same general effect by operating through the pixil buffer. It takes a bit more space on disk, but may be easier to use.

The direct page has two parameters for the address of the start of the pixil buffer (at Q+33), and the end of the pixil buffer (at Q+35), and these can be set to point to anywhere in memory. If changed the alteration must be made after the routine is initialised.

For example, if you open a direct access file with a length of 320 bytes, and poke Hex 989 into Q+33 and poke Hex 989 + 320 (decimal) into Q+35, then the pixil buffer will occupy the same memory as the disk file buffer.

You can then extract each line of pixils from the graphic screen and put it directly to disk as pixils.

At a later stage you can then set the pixil buffer pointers to the same address, and read the pixils back from disk and execute the second part of the dump directly from there.

You can use the same approach as outlined above for combining pictures across the page, except that the result file becomes the actual pixil buffer, rather than a line of pixils across the screen. Hence you can execute the second part of the dump directly after each row of the result file is built up.

If you think you might want to repeat the dump a number of times you can also save the actual print line to be printed into another disk file, and simply print off this file for subsequent dumps.

Each line to be printed for the dump is set up in the string A$ by the ML routine, and this is then printed by Basic. If you open a fifth file, fielded as ...

```
FIELD #5, 243 AS AX$
```

... you can replace the PRINT#-2 command with LSET AX$=A$: PUT #5,P: P=P+1 to produce a direct access file containing all the actual strings required to produce the dump.

You can then repeat the dump at any time, with the following small Basic program

```
OPEN "D",#1,"name",243
FIELD #1, 243 AS A$
FOR X = 1 TO 192
GET #1,X: PRINT #-2,A$;
NEXT X: CLOSE: STOP
```

There are some minor logistical problems with doing 14 badges in this way. Associated with the capacity of each floppy disk. A single floppy will hold 3 badge files plus the file of the print strings for these three badges combined. To do 14 badges therefore requires 5 separate floppy disks to hold the lot.

Setting up the Program

A couple of other programs have to be run before you are able to do any dumps.

"SETINKML/BAS" is a Basic program which sets up the ML code for the dump. It consists mainly of DATA statements which are POKE'd into protected memory.

After running the program, the ML code can be saved, although it is required in memory for the next program.

"SETUP/BAS" is the Basic program used to define the color patterns to be available for dumps.

Run this program, and when you have enough colour patterns defined, save the ML code with the pattern definitions with

```
(C)SAVEM,"INKML",26000,&H7FFF,0
```

The ML routine, INKML, must be loaded into protected memory and be available every time INKDUMP is run.

"INKDUMP/BAS" is then available to be used. Requires a CLEAR 200, 26000 before the (C)LOADM

To define more, or different patterns, run SETUP again and save a new version of INKML.

The other programs, "SAVCOCO3/BAS" and "SAVPMODE/BAS" can be used to save a graphic screen to a direct access disk file.

Listing 1

```
1 '** SETUP - FOR COLOR SCREEN
      DUMP TO CGP-220 PRINTER
      BY  GEORGE MCLINTOCK
2 GOTO 10
3 SAVE"318D:3":END'9
4 'PROGRAM USED TO SETUP COLOR
DEFINITION TABLES FOR INKDUMP
5 ' AND TO PRINT OUT THE DEFINIT
IONS
10 '2 COLOR
20 ' 0 THIS SCREEN PIXIL
30 ' 1 START COLOR
40 ' 2 THIS COLOR
50 ' 3 COL 1
60 ' 4 COL2
70 '
80 ' MULTI COLOR
90 ' 0 THIS SCREEN PIXIL
100 ' 1 START COLOR
110 ' 2 ROLL START  A1
120 ' 3 SWITCH FOR UP/DOWN
130 ' 4 THIS COLOR
140 ' 5 NO OF COLORS  A2
150 ' 6 & 7 START ADDRESS OF COL
ORS
160 '
170 'COLOR BLOCKS
180 ' 0 THIS SCREEN PIXIL
190 ' 1 START COLOR
200 ' 2 ROLL ROW  A1
210 ' 3 SWITCH
220 ' 4 THIS COLOR
230 ' 5 NO OF COLORS  A2
240 ' 6 & 7 START ADDRESS COLORS
250 ' 8 THIS INCREMENT
260 ' 9 ROLL INCREMENT  A3
270 ' 10 INCREMENT EACH INCRE RO
LL  A4
280 '
290 '
300 'ROWS OF COLOR
310 ' 0 THIS SCN PIXIL
```

```
320 ' 1 START COLOR
330 ' 2 ROLL ROW
340 ' 3 ROW COUNT
350 ' 4 THIS COLOR
360 ' 5 NO OF COLORS
370 ' 6&7 START ADDRESS COLORS
380 '
390 'SPACE IS 28 * 2 COLS @ 5 +
25 * ROWS @8 + 100 MULTI @8 + 10
0 BLOCK @ 11
400 ' THEN COLORS AS 25 @ 8 FOR
ROWS + 100 @ 8 FOR MULTI + 100 F
OR BLOCKS @ 11  IE SAME SPACE FO
R COLORS AS FOR PARAMETERS
410 'THE END OF CURRENT COLOR DE
FINITIONS IS IN FIRST 2 BYTES OF
 EACH COLOR BLOCK
420 'MAIN MENU
430 CLS
440 PRINT "SETUP FOR INKDUMP":PR
INT"INKML MUST BE IN MEMORY"
450 PRINT:PRINT "1 SET INITIAL D
EFINITIONS"
460 PRINT "2 DEFINE NEW COLORS"
470 PRINT "3 PRINT COLORS DEFINE
D AS":PRINT"  CHARACTERS"
480 PRINT "4 DUMP COLORS DEFINED
 TO CGP-220  AS THEY WILL APPEAR
"
490 PRINT "ENTER  X  TO FINISH"
500 PRINT:INPUT "ENTER CHOICE";A
$
510 IF A$="X" THEN CLS:PRINT:PRI
NT"RESAVE INKML":PRINT "BY (C)SA
VEM ";CHR$(34);"INKML";CHR$(34);
",26000,&H7FFF,0":STOP
520 IF A$="1" THEN GOSUB 570:GOT
O 440
530 IF A$="2" THEN GOSUB 3460:GO
TO 440
540 IF A$="3" THEN GOSUB 1790:GO
TO 440
550 IF A$="4" THEN GOSUB 2380:GO
TO 440
560 PRINT "INVALID CHOICE":GOTO
390
570 CLS:PRINT "SETTING UP INITIA
L DEFINITIONS"
580 T1=&H6901: T=&H6B14 ' START
OF TWO COLORED PATTERNS
590 Q=T
600 C=8:FOR X=1 TO 7:GOSUB 1610:
NEXT X
610 C=7:FOR X=1 TO 6:GOSUB 1610:
NEXT
620 C=6:FOR X=1 TO 5:GOSUB 1610:
NEXT X
630 C=5:FOR X=1 TO 4:GOSUB 1610:
NEXT X
640 C=4:FOR X=1 TO 3:GOSUB 1610:
NEXT X
650 C=3:FOR X=1 TO 2:GOSUB 1610:
NEXT X
660 C=2:X=1:GOSUB 1610:POKE T1,2
8
670 '
680 ' ROW COLORS ARE HERE IN TAB
LES
690 '
700 GOSUB 1470 'SET FOR ROWS
710 'MULTI COLORED PATTERNS
720 M=R+8*25:M1=&H6906:POKE M1,0
:CX=R+2100+200
730 Q=M:GX=CX:CX=CX+2
740 A1=8:A2=8:GOSUB 1650:POKE M1
,PEEK(M1)+2
750 FOR X=1 TO 8:POKE CX,X:CX=CX
+1:NEXT X
760 'DO RYGY
770 A1=4:A2=4:GOSUB 1650
780 A1=3:GOSUB 1650:POKE M1,PEEK
(M1)+4
790 POKE CX,1:POKE CX+1,4:POKE C
X+2,2:POKE CX+3,4:CX=CX+4
800 'DO RMBM
810 A1=4:A2=4:GOSUB 1650
820 A1=3:GOSUB 1650:POKE M1,PEEK
(M1)+4
830 POKE CX,1:POKE CX+1,5:POKE C
X+2,3:POKE CX+3,5:CX=CX+4
840 'DO GVBV
850 A1=4:A2=4:GOSUB 1650
860 A1=3:GOSUB 1650:POKE M1,PEEK
(M1)+4
870 POKE CX,2:POKE CX+1,6:POKE C
X+2,3:POKE CX+3,6:CX=CX+4
880 'DO SCATTERED PATTERNS
890 A1=8:A2=8:GOSUB 1650
900 A1=4:GOSUB 1650:POKE M1,PEEK
(M1)+4
910 POKE CX,1:POKE CX+1,8:POKE C
X+2,4:POKE CX+3,8:POKE CX+4,2:PO
KE CX+5,8:POKE CX+6,4:POKE CX+7,
8:CX=CX+8
920 A1=8:A2=8:GOSUB 1650
930 A1=4:GOSUB 1650:POKE M1,PEEK
(M1)+4
940 POKE CX,1:POKE CX+1,8:POKE C
X+2,5:POKE CX+3,8:POKE CX+4,3:PO
KE CX+5,8:POKE CX+6,5:POKE CX+7,
8:CX=CX+8
950 A1=8:A2=8:GOSUB 1650:A1=4:GO
SUB 1650:POKE M1,PEEK(M1)+4
960 POKE CX,2:POKE CX+1,8:POKE C
X+2,6:POKE CX+3,8:POKE CX+4,3:PO
KE CX+5,8:POKE CX+6,6:POKE CX+7,
8:CX=CX+8
970 'DO ALL XWV
980 FOR X=1 TO 7
990 A1=3:A2=3:GOSUB 1650
1000 POKE CX,X:POKE CX+1,8:POKE
CX+2,8
1010 CX=CX+3:POKE M1,PEEK(M1)+2
1020 NEXT X
1030 POKE GX,INT(CX/256):POKE GX
+1,CX-INT(CX/256)*256
1040 '
1050 'DO COLOR BLOCKS
1060 B=M+800:B1=&H690B:BE=&H6912
1070 CX=M+1900+800:Q=B:GX=CX:CX=
CX+2
1080 '
1090 A1=32:A2=32:A3=32:A4=0:GOSU
B 1730
1100 A1=8:A2=8:A3=8:A4=0:GOSUB 1
730
1110 A1=12:A2=32:A3=4:A4=4:GOSUB
 1710
1120 FOR X=1 TO 8:FOR Y=1 TO 4
1130 POKE CX,X:CX=CX+1:NEXT Y,X:
POKE B1,4
1140 '
1150 A1=12:A2=16:A3=16:A4=0:GOSU
B 1730
1160 A3=2:A4=2:GOSUB 1710
1170 FOR X=1 TO 8:FOR Y=1 TO 2
1180 POKE CX,X:CX=CX+1
1190 NEXT Y,X:POKE B1,PEEK(B1)+3
1200 '
1210 FOR X=1 TO 7
1220 A1=4:A2=4:A3=2:A4=2:GOSUB 1
730
1230 POKE CX,X:POKE CX+1,X:POKE
CX+2,8:POKE CX+3,8
1240 CX=CX+4:POKE B1,PEEK(B1)+1
1250 NEXT X
1260 FOR X=1 TO 5:GOSUB 1730
1270 POKE CX,2:POKE CX+1,2:POKE
CX+2,X+2:POKE CX+3,X+2
1280 CX=CX+4:POKE B1,PEEK(B1)+1
1290 NEXT X
1300 '
1310 FOR X = 1 TO 7
1320 A1=6:A2=6:A3=3:A4=3:GOSUB 1
730
1330 POKE CX,X:POKE CX+1,X:POKE
CX+2,X:POKE CX+3,8:POKE CX+4,8:P
OKE CX+5,8
1340 CX=CX+6:POKE B1,PEEK(B1)+1
1350 NEXT X
1360 FOR Y=1 TO 6
1370 FOR X=Y TO 6:GOSUB 1730
1380 POKE CX,Y:POKE CX+1,Y:POKE
CX+2,Y:POKE CX+3,X+1:POKE CX+4,X
+1:POKE CX+5,X+1
1390 CX=CX+6:POKE B1,PEEK(B1)+1
1400 NEXT X,Y
1410 '
1420 POKE GX,INT(CX/256):POKE GX
+1,CX-INT(CX/256)*256
1430 '
1440 RETURN
1450 '
1460 'DO ROW COLORS
1470 R=T+5*28:R1=&H6910:POKE R1,
0:CX=T+2100
1480 Q=R:GX=CX:CX=CX+2
1490 A1=3:A2=3:GOSUB 1590
1500 A1=6:GOSUB 1590:A1=3:A2=6:G
OSUB 1590
1510 FOR X=1 TO 6:POKE CX+X-1,X:
NEXT X:  CX=CX+6
1520 A1=4:A2=3:GOSUB 1590:A2=4:G
OSUB 1590
1530 A1=2:A2=6:GOSUB 1590:A2=2:G
OSUB 1590
1540 A1=8:A2=2:GOSUB 1590
1550 POKE CX,4:POKE CX+1,5:POKE
CX+2,6:POKE CX+3,1:POKE CX+4,2:P
OKE CX+5,3:CX=CX+6
1560 POKE R1,8
1570 POKE GX,INT(CX/256):POKE GX
+1,CX-INT(CX/256)*256
1580 RETURN
1590 GOSUB 1670:Q=Q+8:RETURN
1600 'SET 2 COLOR PATTERNS
1610 POKE Q,0:POKE Q+1,0:POKE Q+
2,0:POKE Q+3,X:POKE Q+4,C
1620 Q=Q+5:RETURN
1630 '
1640 'SET MULTI COLORS
1650 GOSUB 1670:Q=Q+8:GOSUB 1670
:POKE Q+3,1:Q=Q+8 ' BOTH SWITCHE
```

```
S
1660 RETURN
1670 POKE Q,0:POKE Q+1,0:POKE Q+
2,A1:POKE Q+3,0:POKE Q+4,0:POKE
Q+5,A2
1680 POKE Q+6,INT(CX/256):POKE Q
+7,CX-INT(CX/256)*256
1690 RETURN
1700 '
1710 GOSUB 1750:Q=Q+11:GOSUB 175
0:POKE Q+3,1:Q=Q+11
1720 RETURN
1730 GOSUB 1750:Q=Q+11:RETURN
1740 'SET COLOR BLOCKS
1750 GOSUB 1670 'SAME
1760 POKE Q+8,0:POKE Q+9,A3:POKE
 Q+10,A4
1770 RETURN
1780 'PRINT OUT WHATS THERE
1790 CLS:PRINT "PRINTING COLORS
DEFINED AS CHARS"
1800 ST=&H68FE: POKE 150,87 ' AT
 600 BAUD
1810 T1=PEEK(ST+3):T=&H6B14
1820 R1=PEEK(ST+18):R=T+5*28
1830 M1=PEEK(ST+8):M=R+8*25
1840 B1=PEEK(ST+13):B=M+8*100
1850 '
1860 PRINT#-2,"TABLE AT ";HEX$(S
T);" - TWO COLORS START AT ";HEX
$(T);" NUMBER = 28"
1870 A=T:FOR X=1 TO 28
1880 PRINT#-2,"T";X;" *";
1890 FOR Y=1 TO 5:PRINT#-2,PEEK(
A);:A=A+1:NEXT Y
1900 PRINT#-2:NEXT X
1910 PRINT#-2,"STOPS AT ";HEX$(A
)
1920 '
1930 PRINT#-2,"MULTI COLORS STAR
T AT ";HEX$(M);"  NUMBER =";M1
1940 A=M:A1=1
1950 FOR WX=1 TO M1
1960 PRINT#-2,"M";A1;" *";
1970 GOSUB 2270:GOSUB 2340
1980 PRINT#-2
1990 A1=A1+1
2000 NEXT WX
2010 PRINT#-2,"STOPS AT ";HEX$(A
)
2020 '
2030 PRINT#-2,"COLOR BLOCKS STAR
T AT ";HEX$(B); "  NUMBER =";B1
2040 A=B:A2=1
2050 FOR WX=1 TO B1
2060 PRINT#-2,"B";A2;" *";
2070 GOSUB 2270
2080 FOR Y=1 TO 3
2090 PRINT#-2,PEEK(A);:A=A+1
2100 NEXT Y:GOSUB 2340
2110 PRINT#-2
2120 A2=A2+1
2130 NEXT WX
2140 PRINT#-2,"STOPS AT ";HEX$(A
)
2150 '
2160 PRINT#-2,"ROW COLORS START
AT ";HEX$(R);"  NUMBER =";R1
2170 A=R:A3=1
2180 FOR WX=1 TO R1
2190 PRINT#-2,"R";A3;" *";
2200 GOSUB 2270:GOSUB 2340
2210 PRINT#-2
2220 A3=A3+1
2230 NEXT WX
2240 PRINT#-2,"STOPS AT ";HEX$(A
)
2250 STOP
2260 '
2270 FOR Y = 0 TO 5
2280 PRINT#-2,PEEK(A);:A=A+1
2290 NEXT Y
2300 D1=PEEK(A-1):D=PEEK(A)*256+
PEEK(A+1):A=A+2
2310 PRINT#-2," ";HEX$(D);
2320 RETURN
2330 '
2340 PRINT#-2," *";
2350 FOR Y=1 TO D1
2360 PRINT#-2,PEEK(D);:D=D+1
2370 NEXT Y:RETURN
2380 CLS:PRINT "DUMPING COLORS T
O CGP-220"
2390 'SET UP PIX BUFFER
2400 PMODE 0,1:PCLS:P=PEEK(&HB7)
*256
2410 POKE 150,18 'AT 2400 BAUD
2420 FOR X=47 TO 58:POKE P+X,0:N
EXT X
2430 POKE &H68F6,2:POKE &H68F7,1
28 'SIZE PIX BUFFER
2440 DEFUSR0=&H68AE:A$=" ":A$=US
R(A$):Q2=&H66AD
2450 P1=PEEK(P+33)*256+PEEK(P+34
)
2460 FOR X=0 TO 7:FOR Y=1 TO 80
2470 POKE P1,X:P1=P1+1 ' SET PIX
ILS IN BUFFER
2480 NEXT Y,X
2490 'NORMAL COLORS
2500 FOR X = 0 TO 7
2510 POKE P+X*2,X: POKE P+X*2+1,
X+1
2520 NEXT X
2530 T3=76:M3=81:B3=86:R3=91
2540 POKE P+T3,0:POKE P+M3,0:POK
E P+B3,0:POKE P+R3,0
2550 BP=&H665E:POKE Q2,8
2560 '
2570 FOR J=1 TO 20
2580 EXEC BP: PRINT#-2,A$;
2590 NEXT J
2600 B$= " ":CY=1:GOSUB 3320
2610 FOR X=0 TO 31:POKE P+X,255:
NEXT X 'EXCLUDE
2620 POKE Q2,0
2630 '
2640 'TWO COLORED PATTERNS
2650 T=&H6B14: TM=&H6914:Q=0:T2=
P+T3: POKE T2,8:Q7=T2:GOSUB 3380
2660 FOR X=1 TO Q8: P1=40: IF X=
Q8 THEN GOSUB 3400
2670 Q1=0
2680 FOR Y=1 TO P1
2690 POKE TM+Q1,PEEK(T+Q):Q=Q+1:
Q1=Q1+1
2700 NEXT Y
2710 R=0:FOR Y=TM TO TM+40 STEP
5
2720 POKE Y,R: R=R+1
2730 NEXT Y
2740 FOR J=1 TO 20
2750 EXEC BP: PRINT#-2,A$;
2760 NEXT J
2770 B$="T":CY=(X-1)*8+1:GOSUB 3
320
2780 NEXT X
2790 '
2800 'MULTI COLORED PATTERNS
2810 RX=T+5*28:M=RX+8*25:MM=&H69
E4:M2=P+M3:Q=0:POKE T2,0:POKE M2
,8
2820 POKE Q2,0:Q7=M2:GOSUB 3380
2830 FOR X=0 TO 31:POKE P+X,255:
NEXT X
2840 FOR X=1 TO Q8:IF X=Q8 THEN
GOSUB 3400
2850 Q1=0
2860 FOR Y=1 TO 64
2870 POKE MM+Q1,PEEK(M+Q):Q=Q+1:
Q1=Q1+1
2880 NEXT Y
2890 R=0:FOR Y=MM TO MM+8*8 STEP
 8
2900 POKE Y,R: R=R+1
2910 NEXT Y
2920 FOR J=1 TO 20
2930 EXEC BP: PRINT#-2,A$;
2940 NEXT J
2950 B$="M":CY=(X-1)*8+1:GOSUB 3
320
2960 NEXT X
2970 '
2980 'COLORED BLOCKS
2990 B=M+800:BM=&H6A64:B2=P+B3:Q
=0:POKE B2,8: POKE M2,0:POKE Q2,
0:Q7=B2:GOSUB 3380
3000 FOR X=1 TO Q8:P1=88:IF X=Q8
 THEN GOSUB 3400
3010 Q1=0
3020 FOR Y=1 TO P1
3030 POKE BM+Q1,PEEK(B+Q):Q=Q+1:
Q1=Q1+1
3040 NEXT Y
3050 R=0:FOR Y=BM TO BM+8*11 STE
P 11
3060 POKE Y,R: R=R+1
3070 NEXT Y
3080 FOR J = 1 TO 20
3090 EXEC BP: PRINT#-2,A$;
3100 NEXT J
3110 B$="B": CY=(X-1)*8+1
3120 GOSUB 3320
3130 NEXT X
3140 '
3150 RM=&H6964:R2=P+R3:Q=0:POKE
B2,0:POKE R2,8:POKE Q2,0:GOSUB 3
380
3160 FOR X=0 TO 31:POKE P+X,255:
NEXT X
3170 FOR X=1 TO Q8:IF X=Q8 THEN
GOSUB 3400
3180 Q1=0: FOR Y=1 TO 64
3190 POKE RM+Q1,PEEK(RX+Q): Q=Q+
1: Q1=Q1+1
3200 NEXT Y
3210 R=0:FOR Y=RM TO RM+8*8 STEP
 8
3220 POKE Y,R: R=R+1
3230 NEXT Y
3240 FOR J=1 TO 20
3250 EXEC BP:PRINT#-2,A$;
3260 NEXT J
3270 B$="R":CY=1:GOSUB 3320
```

```
3280 NEXT X
3290 '
3300 RETURN 'TO MENU
3310 '
3320 PRINT#-2:FOR CX=0 TO 7
3330 PRINT#-2,TAB(CX*11 + 4);
3340 PRINT#-2,B$;CY+CX;
3350 NEXT CX
3360 PRINT#-2
3370 RETURN
3380 Q9=PEEK(Q7+3):Q8=INT(Q9/8):
IF Q9<>Q8*8 THEN Q8=Q8+1
3390 RETURN
3400 IF Q8*8=Q9 THEN RETURN
3410 Q6=Q9-(Q8-1)*8: POKE Q7,Q6:
 POKE Q2,8-Q6
3420 R=0:FOR Y = Q6+1 TO 8
3430 POKE P+R,Y:POKE P+R+1,8: R=
R+2
3440 NEXT Y:RETURN
3450 'DEFINE NEW COLORS
3460 T=&H6B14
3470 R=T+5*28:R3=PEEK(&H690D+3):
X1=(R3-1)*8
3480 R0=T+2100:R2=PEEK(R0)*256+P
EEK(R0+1)
3490 M=R+8*25:M3=PEEK(&H6903+3):
X2=(M3-1)*8
3500 M0=R+2100+200:M2=PEEK(M0)*2
56+PEEK(M0+1)
3510 B=M+8*100:B3=PEEK(&H6908+3)
:X3=(B3-1)*11
3520 B0=M+1900+800: B2=PEEK(B0)*
256 + PEEK(B0+1)
3530 R=R+X1:M=M+X2:B=B+X3
3540 PRINT:PRINT "DEFINE NEW COL
ORS":PRINT "ENTER  X  TO EXIT  O
R"
3550 INPUT "R/M/B";A$: PRINT
3560 IF A$="X" THEN 3620
3570 IF A$="R" THEN CX=R2:Q=R:Y=
R3:GOSUB 3670:R2=CX:R=Q:R3=Y:GOT
O 3540
3580 IF A$="M" THEN CX=M2:Q=M:Y=
M3:GOSUB 3670:M2=CX:M=Q:M3=Y:GOT
O 3540
3590 IF A$="B" THEN CX=B2:Q=B:Y=
B3:GOSUB 3670:B2=CX:B=Q:B3=Y:GOT
O 3540
3600 PRINT:PRINT "INVALID":GOTO
3540
3610 'RESET & EXIT
3620 POKE &H690D+3,R3:POKE R0,IN
T(R2/256):POKE R0+1,R2-INT(R2/25
6)*256
3630 POKE &H6903+3,M3:POKE M0,IN
T(M2/256):POKE M0+1,M2-INT(M2/25
6)*256
3640 POKE &H6908+3,B3:POKE B0,IN
T(B2/256):POKE B0+1,B2-INT(B2/25
6)*256
3650 RETURN 'TO MENU
3660 'SET PATTERN
3670 PRINT:PRINT "DEFINING ";A$;
" PATTERN"
3680 PRINT "NUMBER ALREADY DEFIN
ED IS";Y
3690 PRINT:PRINT"PRESS ENTER TO
CONTINUE":INPUT "OR ENTER X TO A
BORT";B$:IF B$="X" THEN RETURN
3700 Y=Y+1
3710 PRINT:INPUT "ENTER VALUE FO
R ROLL ROW  OR    START";A1
3720 INPUT "ENTER NUMBER OF COLO
RS";A2
3730 IF A$="B" THEN INPUT "ENTER
 ROLL INCERMENT";A3:INPUT "ENTER
 INCREMENT EACH ROLL";A4
3740 IF A$<>"R" THEN INPUT "ENTE
R UP/DOWN SWITCH";X
3750 GOSUB 1670:IF A$<>"R" THEN
POKE Q+3,X
3760 IF A$="B" THEN GOSUB 1760:Q
=Q+11 ELSE Q=Q+8
3770 FOR A4 = 1 TO A2
3780 PRINT "ENTER COLOR NO";A4;
3790 INPUT X:POKE CX,X:CX=CX+1
3800 NEXT A4
3810 RETURN
```

Listing 2

```
                00100 * GALLED INKDMP - SCREEN DUMP FOR INK JET PRINTER
                00110 *
                00130 * USE DIRECT PAGE FOR DATA 6590
                00140         ORG 26000
0000            00150 COLS    EQU     0       COLOR TABLE
0021            00160 SP      EQU     33      ST PIX BUFF
0023            00170 EP      EQU     35      END PIX BUF
0025            00180 STR     EQU     37      START PNT BUFFER
0027            00190 ETR     EQU     39      END PNT BUFF
0029            00200 RED     EQU     41      BYTE FOR RED PIXIL
002B            00210 GREEN   EQU     43      GREEN PIX
002D            00220 BLUE    EQU     45      BYTE FOR BLUE
002F            00230 LMC     EQU     47      LM COLOR
0030            00240 LMN     EQU     48      LM NUM
0031            00250 HLMN    EQU     49      LM > 256
0032            00260 RMC     EQU     50      RM COL
0033            00270 RMN     EQU     51      RM NUM PIX
0034            00280 HRMN    EQU     52      >256
0035            00290 LMBC    EQU     53      LM BORDER COL
0036            00300 LMBN    EQU     54      NUM
0037            00310 RMBC    EQU     55      RM BORDER
0038            00320 RMBN    EQU     56      NUM
0039            00330 TBC     EQU     57      TOP BORDER COLOR
003A            00340 BBC     EQU     58      BOTTOM BORDER
003B            00350 STPM    EQU     59      THIS LINE PMODE
003D            00360 BCON    EQU     61      CNT PIX IN BYTE
003E            00370 THSONE  EQU     62      FOR ALT METHOD
003F            00380 NUMREP  EQU     63      NUM REPEAT EACH PIX
0040            00390 PBYTE   EQU     64      PIXILS PER BYTE
0041            00400 BITPIX  EQU     65      BITS PER PIXIL
```

```
65C5 26  71      01000          BNE     BOTH
65C7 4C          01010          INCA
65C8 A1  02      01020          CMPA    2, X      ROLL START
65CA 2D  01      01030          BLT     NL2
65CC 4F          01040          CLRA              RESET
65CD A7  01      01050 NL2      STA     1, X      NEXT START
65CF 30  08      01060 NLE      LEAX    8, X      NEXT ENTRY
65D1 5A          01070          DECB
65D2 26  EB      01080          BNE     NLLP1
                 01090 *BLOCKS OF COLOR
65D4 D6  56      01100 SBCP     LDB     <BCPN     NUM
65D6 27  26      01110          BEQ     NLOUT     NONE
65D8 9E  57      01120          LDX     <BCPS
65DA A6  01      01130 SBCP1    LDA     1, X      RESET TO START
65DC A7  04      01140          STA     4, X
65DE A6  08      01150          LDA     8, X      THIS ROW CNT
65E0 4C          01160          INCA
65E1 A7  08      01170          STA     8, X      IF CORRECT
65E3 A1  09      01180          CMPA    9, X      ROLL INC
65E5 2D  12      01190          BLT     NLE1
65E7 6F  08      01200          CLR     8, X      ROW CNT
65E9 A6  01      01210          LDA     1, X      OLD START COL
65EB 6D  03      01220          TST     3, X      SWITCH
65ED 26  59      01230          BNE     BOTH1
65EF AB  0A      01240          ADDA    10, X     INC COL VAL
65F1 A7  01      01250          STA     1, X      NEW COL
65F3 A1  02      01260          CMPA    2, X      ROLL ROW
65F5 2D  02      01270          BLT     NLE1
65F7 6F  01      01280          CLR     1, X
65F9 30  0B      01290 NLE1     LEAX    11, X
65FB 5A          01300          DECB
65FC 26  DC      01310          BNE     SBCP1
                 01320 *
                 01330 *ROWS OF COLOR
                 01340 *
65FE D6  5B      01350 NLOUT    LDB     <RCPN
6600 27  21      01360          BEQ     NLOUTX    NONE
6602 9E  5C      01370          LDX     <RCPS
6604 A6  01      01380 NLOW     LDA     1, X      RESET START
6606 A7  04      01390          STA     4, X
6608 A6  03      01400          LDA     3, X      INC ROW CNT
660A 4C          01410          INCA
660B A7  03      01420          STA     3, X
660D A1  02      01430          CMPA    2, X
660F 2D  0D      01440          BLT     NLOX
6611 6F  03      01450          CLR     3, X      CHANGE ROW COL
6613 A6  01      01460          LDA     1, X
6615 4C          01470          INCA
6616 A7  01      01480          STA     1, X
6618 A1  05      01490          CMPA    5, X
661A 2D  02      01500          BLT     NLOX
661C 6F  01      01510          CLR     1, X      RESET COLS
661E 30  08      01520 NLOX     LEAX    8, X
6620 5A          01530          DECB
6621 26  E1      01540          BNE     NLOW
                 01550 *
                 01560 *RESET COLOR POINTERS
                 01570 *
6623 0F  2A      01580 NLOUTX   CLR     <RED+1
```

```
0042                    00410 CNT1    EQU     66
0043                    00420 CNT2    EQU     67
0044                    00430 CNT3    EQU     68
0045                    00440 CNT4    EQU     69
0046                    00450 NBSCN   EQU     70      BYTES ACROSS SCN
0047                    00460 NBLM    EQU     71      BYTES LM SCN
0048                    00470 NBTT    EQU     72      BYTES TO TAKE
0049                    00480 NBRM    EQU     73      BYTES RM SCN
004A                    00490 PIXTAB  EQU     74      ST SCN PIX TABLE
004C                    00500 TCPN    EQU     76      NO 2 COL PAT
004D                    00510 TCPS    EQU     77      ADDR START
004F                    00520 TCPE    EQU     79      END
0051                    00530 MCPN    EQU     81      NUM MULTI
0052                    00540 MCPS    EQU     82      START
0054                    00550 MCPE    EQU     84      END
0056                    00560 BCPN    EQU     86      COL BLOCKS
0057                    00570 BCPS    EQU     87      ST
0059                    00580 BCPE    EQU     89      END
005B                    00590 RCPN    EQU     91
005C                    00600 RCPS    EQU     92
005E                    00610 RCPE    EQU     94
0060                    00620 SCBT    EQU     96      START BLOCK MEM
0062                    00630 SHBT    EQU     98      START BLOCK HERE
                        00640 *
                        00650 *SET PIXIL BUFFER FOR BORDERS TOP & BOTTOM
                        00660 *
6590 96   B7            00670         LDA     <$B7
6592 1F   8B            00680         TFR     A, DP
6594 9E   21            00690         LDX     <SP
6596 96   39            00700         LDA     <TBC
6598 A7   80            00710 BORD1   STA     ,X+
659A 9C   23            00720         CMPX    <EP
659C 26   FA            00730         BNE     BORD1
659E 4F                 00740         CLRA
659F 1F   8B            00750         TFR     A,DP
65A1 39                 00760         RTS
                        00770 *
                        00780 * UPDATE PATTERNS FOR EACH LINE
                        00790 * AT START OF EACH LINE PRINTED
                        00800 *
65A2 D6   4C            00810 NEWLIN  LDB     <TCPN   TWO COL PATTERN
65A4 27   13            00820         BEQ     SMCP    NONE
65A6 9E   4D            00830         LDX     <TCPS
65A8 A6   01            00840 NL0     LDA     1,X     SET THIS START
65AA A7   02            00850         STA     2,X
65AC 27   04            00860         BEQ     NL1
65AE 6F   01            00870         CLR     1, X
65B0 20   02            00880         BRA     NL15
65B2 6C   01            00890 NL1     INC     1, X
65B4 30   05            00900 NL15    LEAX    5, X     NEXT ONE
65B6 5A                 00910         DECB
65B7 26   EF            00920         BNE     NL0
                        00930 * MULTI COL PATTERNS
65B9 D6   51            00940 SMCP    LDB     <MCPN
65BB 27   17            00950         BEQ     SBCP    NONE
65BD 9E   52            00960         LDX     <MCPS
65BF A6   01            00970 NLLP1   LDA     1, X     START
65C1 A7   04            00980         STA     4, X     THIS ONE
65C3 6D   03            00990         TST     3, X     SWITCH
```

```
6625 86   50      01590          LDA     #80
6627 97   2C      01600          STA     <GREEN+1
6629 86   A0      01610          LDA     #160
662B 97   2E      01620          STA     <BLUE+1
                  01630 *CLEAR PNT BUFF
662D 9E   25      01640          LDX     <STR
662F 6F   80      01650 CLRPNT   CLR     , X+
6631 9C   27      01660          CMPX    <ETR
6633 26   FA      01670          BNE     CLRPNT
6635 0F   3E      01680          CLR     <THSONE
6637 39           01690          RTS
                  01700 *
                  01710 *MULTI COL UP AND DOWN
6638 2D   09      01720 BOTH     BLT     DOWN
663A 4C           01730          INCA
663B A1   02      01740          CMPA    2, X     ROLL START
663D 2D   8E      01750          BLT     NL2      NOT YET
663F 63   03      01760 BOTHE    COM     3, X     CHANGE SW
6641 20   8A      01770          BRA     NL2
6643 4A           01780 DOWN     DECA
6644 2E   87      01790          BGT     NL2
6646 20   F7      01800          BRA     BOTHE
                  01810 *BLOCKS UP & DOWN
6648 2D   0C      01820 BOTH1    BLT     DOWN1
664A AB   0A      01830          ADDA    10, X
664C A7   01      01840          STA     1, X
664E A1   02      01850          CMPA    2, X
6650 2D   A7      01860          BLT     NLE1
6652 63   03      01870 BOTH1E   COM     3, X
6654 20   A3      01880          BRA     NLE1
6656 A0   0A      01890 DOWN1    SUBA    10, X
6658 A7   01      01900          STA     1, X
665A 2E   9D      01910          BGT     NLE1
665C 20   F4      01920          BRA     BOTH1E
                  01930 *
                  01940 *PIXILS IN PIX BUFF MOVE TO PNT BUFF
                  01950 *
665E 96   B7      01960 ZZSTAR   LDA     <$B7     SET DP
6660 1F   8B      01970          TFR     A, DP
6662 17   FF3D    01980          LBSR    NEWLIN   UPDATE LINE START
                  01990 *DO LEFT MARGIN
6665 96   2F      02000          LDA     <LMC
6667 0D   31      02010          TST     <HLMN
6669 27   02      02020          BEQ     MG1
666B 8D   2C      02030          BSR     MGM
666D D6   30      02040 MG1      LDB     <LMN
666F 8D   2C      02050          BSR     MGM2     DO THAT MANY
                  02060 *DO LEFT BORDER
6671 96   35      02070          LDA     <LMBC
6673 D6   36      02080          LDB     <LMBN
6675 8D   26      02090          BSR     MGM2
                  02100 *DO PIXILS FROM BUFFER
6677 109E 21      02110          LDY     <SP
667A A6   A0      02120 ST2      LDA     , Y+
667C 8D   2E      02130          BSR     PNT
667E 109C 23      02140          CMPY    <EP
6681 26   F7      02150          BNE     ST2
                  02160 *DO RIGHT BORDER
6683 96   37      02170          LDA     <RMBC
```

```
6685 D6   38      02180          LDB     <RMBN
6687 8D   14      02190          BSR     MGM2
                  02200 *DO RIGHT MARGIN
6689 96   32      02210          LDA     <RMC
668B 0D   34      02220          TST     <HRMN
668D 27   02      02230          BEQ     MG2
668F 8D   08      02240          BSR     MGM
6691 D6   33      02250 MG2      LDB     <RMN
6693 8D   08      02260          BSR     MGM2
6695 4F           02270          CLRA
6696 1F   8B      02280          TFR     A, DP
6698 39           02290          RTS
                  02300 *SET MARGIN PIXILS
6699 0F   43      02310 MGM      CLR     <CNT2   DO 256
669B 20   04      02320          BRA     MGM1
669D D7   43      02330 MGM2     STB     <CNT2
669F 27   0A      02340          BEQ     MGME    ZERO
66A1 97   45      02350 MGM1     STA     <CNT4
66A3 96   45      02360 MGM3     LDA     <CNT4
66A5 8D   05      02370          BSR     PNT
66A7 0A   43      02380          DEC     <CNT2
66A9 26   F8      02390          BNE     MGM3
66AB 39           02400 MGME     RTS
                  02410 *
                  02420 *WHEN GET HERE HAVE PIXIL COLOR IN A
                  02430 *SEARCH TABLE FOR BASIC PRINTER COLORS
                  02440 *
66AC C6   08      02450 PNT      LDB     #8      NO OF BASIC COLORS
66AE 27   09      02460          BEQ     PNTNON
66B0 9E   4A      02470          LDX     <PIXTAB
66B2 A1   81      02480 PNT0     CMPA    , X++
66B4 27   59      02490          BEQ     SETCOA
66B6 5A           02500          DECB
66B7 26   F9      02510          BNE     PNT0
                  02520 *2 COL SEARCH TABLE
66B9 D6   4C      02530 PNTNON   LDB     <TCPN
66BB 27   0B      02540          BEQ     MCP
66BD 9E   4D      02550          LDX     <TCPS
66BF A1   84      02560 PNT2     CMPA    , X
66C1 27   3E      02570          BEQ     SETTCP
66C3 30   05      02580          LEAX    5, X
66C5 5A           02590          DECB
66C6 26   F7      02600          BNE     PNT2
                  02610 *M COL
66C8 D6   51      02620 MCP      LDB     <MCPN
66CA 27   0B      02630          BEQ     BCP
66CC 9E   52      02640          LDX     <MCPS
66CE A1   84      02650 PNT3     CMPA    , X
66D0 27   35      02660          BEQ     SETMCP
66D2 30   08      02670          LEAX    8, X
66D4 5A           02680          DECB
66D5 26   F7      02690          BNE     PNT3
                  02700 *BLOCKS
66D7 D6   56      02710 BCP      LDB     <BCPN
66D9 27   0B      02720          BEQ     RCP
66DB 9E   57      02730          LDX     <BCPS
66DD A1   84      02740 PNT4     CMPA    , X
66DF 27   26      02750          BEQ     SETMCP
66E1 30   0B      02760          LEAX    11, X
```

```
66D4 5A          02680          DECB
66D5 26  F7      02690          BNE     PNT3
                 02700 *BLOCKS
66D7 D6  56      02710 BCP      LDB     <BCPN
66D9 27  0B      02720          BEQ     RCP
66DB 9E  57      02730          LDX     <BCPS
66DD A1  84      02740 PNT4     CMPA    , X
66DF 27  26      02750          BEQ     SETMCP
66E1 30  0B      02760          LEAX    11, X
66E3 5A          02770          DECB
66E4 26  F7      02780          BNE     PNT4
                 02790 *R COLS
66E6 D6  5B      02800 RCP      LDB     <RCPN
66E8 27  0B      02810          BEQ     NULL
66EA 9E  5C      02820          LDX     <RCPS
66EC A1  84      02830 RCPPNT   CMPA    , X
66EE 27  17      02840          BEQ     SETMCP
66F0 30  08      02850          LEAX    8, X
66F2 5A          02860          DECB
66F3 26  F7      02870          BNE     RCPPNT
66F5 C6  08      02880 NULL     LDB     #8      NOT FOUND
66F7 20  18      02890          BRA     PNTSET
                 02900 *
                 02910 * PNTTAB TO SET EACH COLOR DOT
                 02920 *
66F9     8040    02930 PNTTAB   FDB     $8040
66FB     2010    02940          FDB     $2010
66FD     0804    02950          FDB     $0804
66FF     0201    02960          FDB     $0201
                 02970 *GET BASIC PRINTER COLOR INTO B
6701 E6  02      02980 SETTCP   LDB     2, X    FOR 2 COLS
6703 30  03      02990          LEAX    3, X
6705 20  04      03000          BRA     SETCX
6707 E6  04      03010 SETMCP   LDB     4, X    FOR OTHERS
6709 AE  06      03020          LDX     6, X    START COLS
670B E6  85      03030 SETCX    LDB     B, X    CODE REQUIRED
670D 20  02      03040          BRA     PNTSET
670F E6  1F      03050 SETCOA   LDB     -1, X
                 03060 *GO SET BITS IN PNT BUFFER FOR PRINTER COLORS
6711 30  8C E5   03070 PNTSET   LEAX    PNTTAB, PCR
6714 96  3E      03080          LDA     <THSONE
6716 C0  02      03090          SUBB    #2
6718 2D  45      03100          BLT     SET1
671A 27  31      03110          BEQ     SET2
671C C0  02      03120          SUBB    #2
671E 2D  1B      03130          BLT     SET3
6720 27  23      03140          BEQ     SET4
6722 C0  02      03150          SUBB    #2
6724 2D  31      03160          BLT     SET5
6726 27  0B      03170          BEQ     SET6
6728 5A          03180          DECB
6729 27  3C      03190          BEQ     SET7
                 03200 *
                 03210 *SET BITS IN PNT BUFFER
                 03220 *
672B DE  29      03230 SET8     LDU     <RED    &G&B=WHITE
672D E6  C4      03240          LDB     , U
672F EA  86      03250          ORB     A, X
6731 E7  C4      03260          STB     , U
6733 DE  2B      03270 SET6     LDU     <GREEN  &B=VIOLET
```

```
6735 E6   C4      03280          LDB     , U
6737 EA   86      03290          ORB     A, X
6739 E7   C4      03300          STB     , U
673B DE   2D      03310 SET3     LDU     <BLUE   ONLY
673D E6   C4      03320          LDB     , U
673F EA   86      03330          ORB     A, X
6741 E7   C4      03340          STB     , U
6743 20   22      03350          BRA     SET7
                  03360 *
6745 DE   29      03370 SET4     LDU     <RED    &G=YELLOW
6747 E6   C4      03380          LDB     , U
6749 EA   86      03390          ORB     A, X
674B E7   C4      03400          STB     , U
674D DE   2B      03410 SET2     LDU     <GREEN  ONLY
674F E6   C4      03420          LDB     , U
6751 EA   86      03430          ORB     A, X
6753 E7   C4      03440          STB     , U
6755 20   10      03450          BRA     SET7
                  03460 *
6757 DE   2D      03470 SET5     LDU     <BLUE   &R=MAGENTA
6759 E6   C4      03480          LDB     , U
675B EA   86      03490          ORB     , X
675D E7   C4      03500          STB     , U
675F DE   29      03510 SET1     LDU     <RED
6761 E6   C4      03520          LDB     , U
6763 EA   86      03530          ORB     A, X
6765 E7   C4      03540          STB     , U
6767 4C           03550 SET7     INCA            BLACK & COMMON
6768 81   08      03560          CMPA    #8      EXIT
676A 2D   07      03570          BLT     NOTX1   INC THSONE
676C 4F           03580          CLRA
676D 0C   2A      03590          INC     <RED+1  INCREASE
676F 0C   2C      03600          INC     <GREEN+1 POINTERS
6771 0C   2E      03610          INC     <BLUE+1 TO COLORS
6773 97   3E      03620 NOTX1    STA     <THSONE
                  03630 *
                  03640 *INC COL PATTERNS FOR EACH PIXIL
                  03650 *AFTER EACH DOT IS SET UP
                  03660 *TWO COL PAT
6775 D6   4C      03670          LDB     <TCPN
6777 27   11      03680          BEQ     UMCP
6779 9E   4D      03690          LDX     <TCPS
677B 6D   02      03700 UPTN1    TST     2, X
677D 27   04      03710          BEQ     UPTN2
677F 6F   02      03720          CLR     2, X
6781 20   02      03730          BRA     UPTN3
6783 6C   02      03740 UPTN2    INC     2, X
6785 30   05      03750 UPTN3    LEAX    5, X
6787 5A           03760          DECB
6788 26   F1      03770          BNE     UPTN1
                  03780 *MULTI COL
678A D6   51      03790 UMCP     LDB     <MCPN
678C 27   11      03800          BEQ     UBCP
678E 9E   52      03810          LDX     <MCPS
6790 A6   04      03820 SEXMCP   LDA     4, X    THIS ONE
6792 4C           03830          INCA
6793 A1   05      03840          CMPA    5, X
6795 2D   01      03850          BLT     RS
6797 4F           03860          CLRA
6798 A7   04      03870 RS       STA     4, X
```

```
679A 30   08       03880          LEAX    8, X
679C 5A            03890          DECB
679D 26   F1       03900          BNE     SEXMCP
                   03910 *COL BLOCKS
679F D6   56       03920 UBCP     LDB     <BCPN
67A1 27   11       03930          BEQ     UEXT
67A3 9E   57       03940          LDX     <BCPS
67A5 A6   04       03950 SETBCP   LDA     4, X
67A7 4C            03960          INCA
67A8 A1   05       03970          CMPA    5, X
67AA 2D   01       03980          BLT     RSB
67AC 4F            03990          CLRA
67AD A7   04       04000 RSB      STA     4, X
67AF 30   0B       04010          LEAX    11, X
67B1 5A            04020          DECB
67B2 26   F1       04030          BNE     SETBCP
67B4 39            04040 UEXT     RTS
                   04050 *
                   04060 *
                   04070 * SPACE FOR OTHER ONE
                   04080 *
67C0               04090          ORG     $67C0
                   04100 *
                   04110 * REVERSE PIXILS IN PIX BUFFER - MIRROR IMAGE EFFECT
                   04120 *
67C0 96   B7       04130          LDA     <$B7    SET DIRECT
67C2 1F   8B       04140          TFR     A, DP    PAGE REGISTER
67C4 9E   21       04150          LDX     <SP
67C6 DE   23       04160          LDU     <EP
67C8 A6   84       04170 EXCH1    LDA     , X      EXCHANGE PIXILS
67CA E6   C2       04180          LDB     , -U
67CC A7   C4       04190          STA     , U
67CE E7   80       04200          STB     , X+
67D0 DF   42       04210          STU     <CNT1
67D2 9C   42       04220          CMPX    <CNT1
67D4 2D   F2       04230          BLT     EXCH1
67D6 20   4D       04240          BRA     EXA4
                   04250 *
                   04260 *EXTRACT PIXILS FROM GRAPHIC SCREEN AND PUT IN PIX BUFFER
                   04270 *COCO 3 ENTRY
67D8 10FF 02D2     04280 START3   STS     $2D2    CHANGE STACK
67DC 10CE 02D0     04290          LDS     #$2D0
67E0 34   66       04300          PSHS    A, B, Y, U SAVE REGISTERS
67E2 8D   09       04310          BSR     START
67E4 35   66       04320 Z31E     PULS    A, B, Y, U RESTORE REGISTERS
67E6 10FE 02D2     04330          LDS     $2D2    STACK
67EA 7E   E6EB     04340          JMP     $E6EB   RETURN TO BASIC
                   04350 *
                   04360 *OLD COCO ENTRY
                   04370 *
67ED 96   B7       04380 START    LDA     <$B7    SET DP REG
67EF 1F   8B       04390 STC3     TFR     A, DP
67F1 9E   3B       04400          LDX     <STPM
67F3 D6   47       04410          LDB     <NBLM
67F5 3A            04420          ABX             LM SCN
67F6 96   48       04430          LDA     <NBTT   BYTES TO TAKE
67F8 97   42       04440          STA     <CNT1
67FA DE   21       04450          LDU     <SP     PNT BUFF
                   04460 *
```

```
67FC 96 40      04470 EXA0   LDA   <PBYTE  PIX PER BYTE
67FE 97 43      04480        STA   <CNT2
6800 E6 80      04490        LDB   ,X+     THIS SCN BYTE
6802 96 41      04500 EXA1   LDA   <BITPIX
6804 97 44      04510        STA   <CNT3
6806 4F         04520        CLRA
6807 59         04530 EXA2   ROLB          EXTRACT EACH PIXIL
6808 49         04540        ROLA
6809 0A 44      04550        DEC   <CNT3
680B 26 FA      04560        BNE   EXA2
680D D7 45      04570        STB   <CNT4   SAVE IT
680F D6 3F      04580        LDB   <NUMREP
6811 A7 C0      04590 EXA3   STA   ,U+     REPEAT FOR SIZE
6813 5A         04600        DECB
6814 26 FB      04610        BNE   EXA3
6816 D6 45      04620        LDB   <CNT4   RESTORE IT
6818 0A 43      04630        DEC   <CNT2
681A 26 E6      04640        BNE   EXA1
681C 0A 42      04650        DEC   <CNT1
681E 26 DC      04660        BNE   EXA0
                04670 *
6820 D6 49      04680        LDB   <NBRM
6822 3A         04690 EXA9   ABX           RM SCN
6823 9F 3B      04700        STX   <STPM   START NEXT LINE
                04710 *
6825 4F         04720 EXA4   CLRA          RESTORE DP
6826 1F 8B      04730        TFR   A,DP
6828 39         04740        RTS
                04750 *
                04760 *SKIP LINE ACROSS
                04770 *SAME BOTH COCO'S
6829 96 B7      04780 ASKIP  LDA   <$B7    SET DP
682B 1F 8B      04790 EXA5   TFR   A,DP
682D 9E 3B      04800        LDX   <STPM   INC FOR START
682F D6 46      04810        LDB   <NBSCN  NEXT LINE
6831 20 EF      04820        BRA   EXA9
                04830 *
                04840 *SKIP LINE DOWN PAGE
                04850 *SAME BOTH
6833 96 B7      04860 DSKIP  LDA   <$B7    SET DP
6835 1F 8B      04870        TFR   A,DP
6837 96 3D      04880 DSKIP2 LDA   <BCON
6839 4C         04890        INCA
683A 91 40      04900        CMPA  <PBYTE  INC POINTER
683C 2D 07      04910        BLT   DSKIPE
683E 4F         04920        CLRA
683F 9E 3B      04930        LDX   <STPM
6841 30 01      04940        LEAX  1,X
6843 9F 3B      04950        STX   <STPM
6845 97 3D      04960 DSKIPE STA   <BCON
6847 20 DC      04970        BRA   EXA4
                04980 *
                04990 *EXTRACT PIXILS DOWN SCREEN
                05000 *
6849 96 B7      05010 EXD    LDA   <$B7
684B 1F 8B      05020        TFR   A,DP
684D 9E 3B      05030 EXDC3  LDX   <STPM
684F 96 47      05040        LDA   <NBLM   LM SCN
6851 27 06      05050        BEQ   EXD2    NONE
```

```
6853 D6   46       05060          LDB     <NBSCN
6855 3A            05070 EXD1     ABX
6856 4A            05080          DECA
6857 26   FC       05090          BNE     EXD1
6859 96   48       05100 EXD2     LDA     <NBTT
685B 97   42       05110          STA     <CNT1
685D DE   21       05120          LDU     <SP     PNT BUFF
685F E6   84       05130 EXE1     LDB     , X      THIS SCN BYTE
6861 96   3D       05140          LDA     <BCON   COUNTER
6863 27   0C       05150          BEQ     EXD5    FIRST PIX
6865 97   43       05160          STA     <CNT2
6867 96   41       05170 EXD3     LDA     <BITPIX SKIP THOSE NOT REQUIRED
6869 59            05180 EXD4     ROLB
686A 4A            05190          DECA
686B 26   FC       05200          BNE     EXD4
686D 0A   43       05210          DEC     <CNT2
686F 26   F6       05220          BNE     EXD3
                   05230 *NOW THIS PIXIL IN B
6871 96   41       05240 EXD5     LDA     <BITPIX
6873 97   43       05250          STA     <CNT2
6875 4F            05260          CLRA
6876 59            05270 EXD6     ROLB            EXTRACT THIS PIXIL
6877 49            05280          ROLA
6878 0A   43       05290          DEC     <CNT2
687A 26   FA       05300          BNE     EXD6
687C D6   3F       05310          LDB     <NUMREP
687E A7   C0       05320 EXD7     STA     , U+     SCALE UP
6880 5A            05330          DECB
6881 26   FB       05340          BNE     EXD7
                   05350 *DO ALL PIXILS
6883 D6   46       05360          LDB     <NBSCN
6885 3A            05370          ABX
6886 0A   42       05380          DEC     <CNT1
6888 26   D5       05390          BNE     EXE1
                   05400 *TEST IF DONE BYTE & SET FOR NEXT
688A 20   AB       05410          BRA     DSKIP2
                   05420 *COCO 3 ENTRY EXTRACT DOWN
                   05430 *
688C 10FF 02D2     05440          STS     $2D2    SAVE STACK
6890 10CE 02D0     05450          LDS     #$2D0
6894 34   36       05460          PSHS    A, B, X, Y
6896 8D   B1       05470          BSR     EXD
6898 16   FF49     05480          LBRA    Z31E    RESTORE THINGS
                   05490 *
                   05500 *COCO 3 ENTRY FOR SET UP
                   05510 *
689B 31   8D FF39  05520          LEAY    START3, PCR
689F 5F            05530          CLRB
68A0 DE   BA       05540          LDU     $BA     MOVE EXTRACT PROGRAM
68A2 A6   A0       05550 Z31      LDA     , Y+     TO GRAPHIC SCREEN
68A4 A7   C0       05560          STA     , U+
68A6 5A            05570          DECB
68A7 26   F9       05580          BNE     Z31
                   05590 *
68A9 CE   2000     05600          LDU     #$2000  GRAPHIC SCN START
68AC 20   02       05610          BRA     ZSTAR1
                   05620 *
                   05630 *SET UP POINTERS ETC IN DP FOR DUMP
                   05640 *
68AE DE   BA       05650 ZSTART   LDU     <$BA    START GRAPHIC PAGE
```

```
68B0 DC   B7       05660 ZSTAR1  LDD     <$B7    END GRAPHIC PAGE
68B2 1F   8B       05670         TFR     A, DP    SET DP
68B4 DF   3B       05680         STU     <STPM   START GRAPHIC PAGE
68B6 DD   4A       05690         STD     <PIXTAB PIX TABLE
68B8 C6   4C       05700         LDB     #TCPN
68BA DD   62       05710         STD     <SHBT
68BC 5F            05720         CLRB
68BD 4C            05730         INCA            +256
68BE DD   25       05740         STD     <STR    START PNT BUFF
68C0 1F   03       05750         TFR     D, U
68C2 33   5D       05760         LEAU    -3, U
68C4 DD   29       05770         STD     <RED    IN PNT BUFF
68C6 DD   2B       05780         STD     <GREEN
68C8 DD   2D       05790         STD     <BLUE
68CA C6   F0       05800         LDB     #240
68CC DD   27       05810         STD     <ETR
                   05820 *
68CE EF   02       05830         STU     2, X     SET STRING
68D0 C6   F3       05840         LDB     #243
68D2 E7   84       05850         STB     , X      STR LEN
68D4 CC   1B43     05860         LDD     #$1B43  PRINTER CONTROL CODES
68D7 ED   C4       05870         STD     , U
68D9 86   50       05880         LDA     #80
68DB A7   42       05890         STA     2, U
                   05900 *
68DD 30   8D 001D  05910         LEAX    STTAB, PCR
68E1 9F   60       05920         STX     <SCBT
68E3 0F   3D       05930         CLR     <BCON
68E5 DE   62       05940         LDU     <SHBT   MOVE COL PAT TABLE
68E7 C6   14       05950         LDB     #20
68E9 A6   80       05960 ST1     LDA     , X+
68EB A7   C0       05970         STA     , U+
68ED 5A            05980         DECB
68EE 26   F9       05990         BNE     ST1
                   06000 *
68F0 DC   27       06010         LDD     <ETR
68F2 5C            06020         INCB
68F3 DD   21       06030         STD     <SP
68F5 C3   0100     06040         ADDD    #256
68F8 DD   23       06050         STD     <EP
                   06060 *
68FA 4F            06070         CLRA
68FB 1F   8B       06080         TFR     A, DP
68FD 39            06090         RTS
                   06100 *
                   06110 *SET TABLE POINTERS HERE
                   06120 *
68FE      00       06130 STTAB   FCB     0
68FF      6914     06140         FDB     TWOCOL
6901      0000     06150         FDB     0
6903      00       06160         FCB     0
6904      69E4     06170         FDB     MULTCL
6906      0000     06180         FDB     0
6908      00       06190         FCB     0
6909      6A64     06200         FDB     BLKCOL
690B      0000     06210         FDB     0
690D      00       06220         FCB     0
690E      6964     06230         FDB     ROWCOL
6910      0000     06240         FDB     0
```

```
6912    6B14    06250          FDB     COLORS
                06260 *EXECUTE TIME TABLES
6914            06270 TWOCOL   RMB     16*5
6964            06280 ROWCOL   RMB     16*8
69E4            06290 MULTCL   RMB     16*8
6A64            06300 BLKCOL   RMB     16*11
6B14            06310 COLORS   RMB     256
                06320 *
                06321 *MEMORY STORAGE FOR COLOR DEFINITION
                06323 *START FROM COLORS (HEX 6B14 AS ASSEMBLED)
6C14            06330 ZZENDX   EQU     *
67ED            06340          END     START

00000 TOTAL ERRORS
```

Listing 3

```
1 '**  INKML/BAS
       BY GEORGE MCLINTOCK
       JULY 87
2 GOTO 10
3 SAVE"318E:3":END'9
4 'PROGRAM WHICH SETS UP ML ROUT
INES FOR INKDUMP
10 CLEAR 200,26000
20 M=26000:M1=M
30 LN=3000:FOR X=0 TO 899 STEP 2
5:N = 25
40 GOSUB 90:NEXT X
50 CLS:PRINT "INKML ML ROUTINES
NOW IN MEMORY RUN  SETUP  PROGRA
M TO ESTABLISHCOLOR DEFINITIONS"
60 PRINT "IT CAN THEN BE USED WI
TH INKDUMP"
70 STOP
80 '
90 PRINT LN;:A=0:FOR Y=0 TO N-1
100 READ C$:B=VAL("&H"+C$):A=A+B
:POKE M,B:M=M+1
110 NEXT Y:READ C$:IF A<> VAL("&
H"+C$) THEN PRINT "ERROR IN LINE
 NO";LN:STOP
120 LN=LN+10:RETURN
130 '
3000 DATA 96,B7,1F,8B,9E,21,96,3
9,A7,80,9C,23,26,FA,4F,1F,8B,39,
D6,4C,27,13,9E,4D,A6,AAA
3010 DATA 1,A7,2,27,4,6F,1,20,2,
6C,1,30,5,5A,26,EF,D6,51,27,17,9
E,52,A6,1,A7,71B
3020 DATA 4,6D,3,26,71,4C,A1,2,2
D,1,4F,A7,1,30,8,5A,26,EB,D6,56,
27,26,9E,57,A6,7D6
3030 DATA 1,A7,4,A6,8,4C,A7,8,A1
,9,2D,12,6F,8,A6,1,6D,3,26,59,AB
,A,A7,1,A1,749
3040 DATA 2,2D,2,6F,1,30,B,5A,26
,DC,D6,5B,27,21,9E,5C,A6,1,A7,4,
A6,3,4C,A7,3,79C
3050 DATA A1,2,2D,D,6F,3,A6,1,4C
,A7,1,A1,5,2D,2,6F,1,30,8,5A,26,
E1,F,2A,86,687
3060 DATA 50,97,2C,86,A0,97,2E,9
E,25,6F,80,9C,27,26,FA,F,3E,39,2
D,9,4C,A1,2,2D,8E,8F9
3070 DATA 63,3,20,8A,4A,2E,87,20
,F7,2D,C,AB,A,A7,1,A1,2,2D,A7,63
,3,20,A3,A0,A,806
3080 DATA A7,1,2E,9D,20,F4,96,B7
,1F,8B,17,FF,3D,96,2F,D,31,27,2,
8D,2C,D6,30,8D,2C,975
3090 DATA 96,35,D6,36,8D,26,10,9
E,21,A6,A0,8D,2E,10,9C,23,26,F7,
96,37,D6,38,8D,14,96,A58
3100 DATA 32,D,34,27,2,8D,8,D6,3
3,8D,8,4F,1F,8B,39,F,43,20,4,D7,
43,27,A,97,45,69E
3110 DATA 96,45,8D,5,A,43,26,F8,
39,C6,8,27,9,9E,4A,A1,81,27,59,5
A,26,F9,D6,4C,27,95B
3120 DATA B,9E,4D,A1,84,27,3E,30
,5,5A,26,F7,D6,51,27,B,9E,52,A1,
84,27,35,30,8,5A,888
3130 DATA 26,F7,D6,56,27,B,9E,57
,A1,84,27,26,30,B,5A,26,F7,D6,5B
,27,B,9E,5C,A1,84,A16
3140 DATA 27,17,30,8,5A,26,F7,C6
,8,20,18,80,40,20,10,8,4,2,1,E6,
2,30,3,20,4,531
3150 DATA E6,4,AE,6,E6,85,20,2,E
6,1F,30,8C,E5,96,3E,C0,2,2D,45,2
7,31,C0,2,2D,1B,93B
3160 DATA 27,23,C0,2,2D,31,27,B,
5A,27,3C,DE,29,E6,C4,EA,86,E7,C4
,DE,2B,E6,C4,EA,86,C48
3170 DATA E7,C4,DE,2D,E6,C4,EA,8
6,E7,C4,20,22,DE,29,E6,C4,EA,86,
E7,C4,DE,2B,E6,C4,EA,1126
3180 DATA 86,E7,C4,20,10,DE,2D,E
6,C4,EA,86,E7,C4,DE,29,E6,C4,EA,
86,E7,C4,4C,81,8,2D,EFF
3190 DATA 7,4F,C,2A,C,2C,C,2E,97
,3E,D6,4C,27,11,9E,4D,6D,2,27,4,
6F,2,20,2,6C,5B1
3200 DATA 2,30,5,5A,26,F1,D6,51,
27,11,9E,52,A6,4,4C,A1,5,2D,1,4F
,A7,4,30,8,5A,74D
3210 DATA 26,F1,D6,56,27,11,9E,5
7,A6,4,4C,A1,5,2D,1,4F,A7,4,30,B
,5A,26,F1,39,FF,918
3220 DATA FF,FF,0,0,0,0,FF,FF,FF
,FF,96,B7,1F,8B,9E,21,DE,23,A6,8
4,E6,C2,A7,C4,E7,ED5
3230 DATA 80,DF,42,9C,42,2D,F2,2
0,4D,10,FF,2,D2,10,CE,2,D0,34,66
,8D,9,35,66,10,FE,A77
3240 DATA 2,D2,7E,E6,EB,96,B7,1F
,8B,9E,3B,D6,47,3A,96,48,97,42,D
E,21,96,40,97,43,E6,C96
3250 DATA 80,96,41,97,44,4F,59,4
9,A,44,26,FA,D7,45,D6,3F,A7,C0,5
A,26,FB,D6,45,A,43,B0C
3260 DATA 26,E6,A,42,26,DC,D6,49
,3A,9F,3B,4F,1F,8B,39,96,B7,1F,8
B,9E,3B,D6,46,20,EF,ABA
3270 DATA 96,B7,1F,8B,96,3D,4C,9
1,40,2D,7,4F,9E,3B,30,1,9F,3B,97
,3D,20,DC,96,B7,1F,98A
3280 DATA 8B,9E,3B,96,47,27,6,D6
,46,3A,4A,26,FC,96,48,97,42,DE,2
1,E6,84,96,3D,27,C,A56
3290 DATA 97,43,96,41,59,4A,26,F
C,A,43,26,F6,96,41,97,43,4F,59,4
9,A,43,26,FA,D6,3F,A03
3300 DATA A7,C0,5A,26,FB,D6,46,3
A,A,42,26,D5,20,AB,10,FF,2,D2,10
,CE,2,D0,34,36,8D,AD4
3310 DATA B1,16,FF,49,31,8D,FF,3
9,5F,DE,BA,A6,A0,A7,C0,5A,26,F9,
CE,20,0,20,2,DE,BA,CCA
3320 DATA DC,B7,1F,8B,DF,3B,DD,4
A,C6,4C,DD,62,5F,4C,DD,25,1F,3,3
3,5D,DD,29,DD,2B,DD,C19
3330 DATA 2D,C6,F0,DD,27,EF,2,C6
,F3,E7,84,CC,1B,43,ED,C4,86,50,A
7,42,30,8D,0,1D,9F,D0F
3340 DATA 60,F,3D,DE,62,C6,14,A6
,80,A7,C0,5A,26,F9,DC,27,5C,DD,2
1,C3,1,0,DD,23,4F,B3C
3350 DATA 1F,8B,39,0,69,14,0,0,0
,69,E4,28,0,0,6A,64,30,0,0,69,64
,8,0,6B,14,527
```

Listing 4

```
1 '** INKDUMP TO CGP-220
      BY  GEORGE MCLINTOCK
      JULY 1987
2 GOTO 10
3 SAVE"318C:3":END'9
4 'PROGRAM PERFORMS A COLOR SCRE
EN DUMP FROM ANY COCO GRAPHIC SC
REEN TO CGP-220 INK JET PRINTER
5 'ML ROUTINE CALLED INKML MUST
BE IN MEMORY FOR IT TO WORK
6 'OTHER PROGRAM CALLED SETUP MU
ST ALSO HAVE BEEN RUN BEFORE
7 ' TO SET UP COLORS IN INKML
8 ' PRINTER BAUD RATE IS SET TO
2400
10 GOTO 320 ' SETUP
20 'TOP MARGIN
30 IF BN(7)>0 THEN POKE Q+57,BM(
7):EXEC Q3:FOR X=1 TO BN(7): EXE
C BP: PRINT#-2,A$;:NEXT X
40 'TOP BORDER
50 IF BN(3)>0 THEN POKE Q+57,BM(
3):EXEC Q3:FOR X=1 TO BN(3):EXEC
 BP:PRINT#-2,A$;:NEXT X
60 'TOP SCREEN MARGIN
70 IF SM(3)>0 THEN FOR X=1 TO SM
(3):EXEC SK:NEXT X
80 'DO DUMP ITSELF FROM SCREEN
90 IF P2=2.5 THEN GOSUB 240:GOTO
 160
100 FOR X=1 TO W
110 IF P1=3 THEN EXEC EX
115 IF P1<>3 THEN HCLS
120 IF IX<>0 THEN EXEC Q4 'REVER
SE IMAGE
130 FOR Y=1 TO P2
140 EXEC BP:PRINT#-2,A$;
150 NEXT Y,X
160 'BOTTOM BORDER
170 IF BN(4)>0 THEN POKE Q+57,BM
(4):EXEC Q3:FOR X=1 TO BN(4):EXE
C BP:PRINT#-2,A$;:NEXT X
180 'BOTTOM MARGIN
190 IF BN(8)>0 THEN POKE Q+57,BM
(8):EXEC Q3:FOR X=1 TO BN(8):EXE
C BP:PRINT#-2,A$;:NEXT X
200 '
210 STOP
220 '
230 '2.5 SIZE PMODE
240 P2=3:FOR X=1 TO W
250 EXEC EX
260 FOR Y=1 TO P2
270 EXEC BP:PRINT#-2,A$;
280 NEXT Y:IF P2=3 THEN P2=2 ELS
E P2=3
290 NEXT X
300 RETURN
310 '
320 'INITIALISE VARIABLES
330 PCLEAR 5: CLEAR 500,26000: P
OKE 150,18
340 DIM A$,X,Y,BP,EX,SK,W
350 DIM P$(16),BM(8),BM$(8),SM(4
),BN(8),BN$(8)
360 CLS:PRINT "SCREEN DUMP TO CG
P-220":PRINT:PRINT"BY GEORGE MCL
INTOCK"
370 FOR X=1 TO 8:BM$(X)="8":BM(X
)=64+X:NEXT X
380 PRINT:PRINT "SELECT GRAPHIC
SCREEN TO DUMP":PRINT "1 HSCREEN
 1":PRINT "2 HSCREEN 2":PRINT "3
 PMODE 3"
390 ND=192:N3=1:INPUT P1
400 IF P1=1 THEN NA=320:NP=4:NB=
2:N1=80
410 IF P1=2 THEN NA=320:NP=2:NB=
4:N1=160
420 IF P1=3 THEN NA=256:NP=4:NB=
2:N1=32:N3=2
430 N2=N1:IF NA=0 THEN 380
440 IF P1=3 THEN PMODE 3,1 ELSE
PMODE 0,1
450 WP=640:GOSUB 1650
460 PRINT:PRINT "ENTER SIZE OF D
UMP":P3=3
470 PRINT "1 SINGLE":PRINT "2 DO
UBLE":PRINT "3 THREE TIMES"
480 IF P1=3 THEN PRINT "4 FULL W
IDTH (2.5) TIMES":P3=4
490 PRINT "ENTER FOR OTHER SELEC
TIONS"
500 INPUT P2: IF P2<0 OR P2>P3 T
HEN 460
510 IF P2=3 THEN E$="R":IX=1 ELS
E IF P2=0 THEN GOSUB 1930 ELSE I
F P2=4 THEN P2=2.5
520 PRINT:INPUT "ENTER BORDER SI
ZE";P3: IF P3 > 0 THEN GOSUB 210
0
530 PRINT:PRINT "PRESS ENTER FOR
 DUMP TO BE":PRINT"CENTERED ON P
AGE":PRINT"WITH WHITE MARGINS"
540 INPUT "OR ANY CHARACTER TO S
ELECT";B$: IF B$ <> "" THEN GOSU
B 2160 ELSE GOSUB 2270
550 PRINT:PRINT "ENTER ANY CHARA
CTER FOR EXTRA   MENU OR PRESS E
NTER TO BYPASS"
560 INPUT C$: IF C$ <> "" THEN G
OSUB 2380
570 '
580 CLS:PRINT "SELECT PRINTER CO
LORS FOR DUMP  ENTER PRINTER COL
ORS BY CODES   FROM SETUP"
590 IF P1=2 THEN NC=16 ELSE NC=4
600 FOR X=0 TO NC-1
610 PRINT "FOR PIXIL";X;:INPUT P
$(X)
620 NEXT X
630 FOR X=0 TO NC-1: FOR Y=1 TO
8 'EQUATE BORDER & PIXIL COLORS
640 IF P$(X) = BM$(Y) THEN BM(Y)
=X
650 NEXT Y,X
660 FOR X=1 TO 7: FOR Y=X+1 TO 8
 'EQUATE MARGINS
670 IF BM$(X)=BM$(Y) THEN BM(Y)=
BM(X)
680 NEXT Y,X
690 '
700 'SET DEFAULTS
710 FOR X=47 TO 58:POKE Q+X,0:NE
XT X
720 POKE Q2,0:POKE T2,0:POKE M2,
0:POKE B2,0:POKE R2,0
730 'SET MARGINS
740 POKE Q+47,BM(5):POKE Q+50,BM
(6) 'MARGIN COLORS
750 IF BN(5)<256 THEN POKE Q+48,
BN(5) ELSE POKE Q+49,1:POKE Q+48
,BN(5)-256
760 IF BN(6)<256 THEN POKE Q+51,
BN(6) ELSE POKE Q+52,1:POKE Q+51
,BN(6)-256
770 'BORDERS
780 POKE Q+53,BM(1):POKE Q+54,BN
(1)
790 POKE Q+55,BM(2):POKE Q+56,BN
(2)
800 'SET TO EXTRACT FROM SCREEN
810 POKE Q+63,P2:POKE Q+64,NP:PO
KE Q+65,NB
820 POKE Q+70,N1:POKE Q+71,SM(1)
/NP:POKE Q+73,SM(2)/NP
830 IF E$="" THEN W=ND:POKE Q+72
,N1-PEEK(Q+71)-PEEK(Q+73) ELSE W
=NA:POKE Q+72,ND
840 'SET COLORS
850 FOR X=0 TO NC-1
860 Y=0:X9=X:IF P$(X)="" THEN Y=
1:GOTO 900
870 IF LEN(P$(X))=1 THEN A$=" ":
F$=P$(X):GOSUB 1200:GOTO 890
880 A$=LEFT$(P$(X),1):F$=MID$(P$
(X),2):GOSUB 1200
890 IF Y=0 THEN 920
900 GOSUB 1150:PRINT "PIXIL";X
910 INPUT F$:IF F$="X" THEN STOP
 ELSE IF F$="N" THEN GOSUB 2310:
GOTO 860 ELSE P$(X)=F$:GOTO 860
920 NEXT X
930 'MARGIN & BORDER
940 FOR X=1 TO 8:IF BM(X)<=16 TH
EN 1040
950 IF X=1 THEN 980
960 FOR Y=1 TO X-1:IF BM(X)=BM(Y
) THEN Y=20
970 NEXT Y: IF Y>10 THEN 1040
980 Y=0:X9=BM(X):IF BM$(X)="" TH
EN Y=1:GOTO 1020
990 IF LEN(BM$(X))=1 THEN A$=" "
:F$=BM$(X):GOSUB 1200:GOTO 1010
1000 A$=LEFT$(BM$(X),1):F$=MID$(
BM$(X),2):GOSUB 1200
1010 IF Y=0 THEN 1040
1020 GOSUB 1150:PRINT BM$(X)
1030 INPUT F$:IF F$="X" THEN STO
P ELSE IF F$="N" THEN GOSUB 2310
:GOTO 980 ELSE BM$(X)=F$:GOTO 98
0
1040 NEXT X
1050 '
1060 IF E$="" THEN X0=NA*P2:W=ND
: ELSE X0=ND*P2:W=NA 'SET WIDTH
PIX BUFFER
1070 POKE Q1,INT(X0/256):POKE Q1
+1,X0-INT(X0/256)*256
1080 'SET CALLS TO EXTRACT
1090 IF P1=3 THEN DEFUSR0=&H68AE
:IF E$="" THEN EX=&H67ED:SK=&H68
29:POKE Q+63,P2*2 ELSE EX=&H6849
:SK=&H6833
1100 IF P1<>3 THEN DEFUSR0=&H689
B:POKE &HE6C6,18:POKE &HE6C7,18:
POKE &HE6E1,&H7E:IF E$="" THEN P
OKE &HE6E2,&H0E:POKE &HE6E3,0:SK
=&H6829 ELSE POKE &HE6E2,&H0E:PO
KE &HE6E3,180:SK=&H6833
1110 A$=USR(A$) ' INITIALISE
1120 IF P1=3 THEN SCREEN 1,1
1125 IF P1 <> 3 THEN HSCREEN P1
```

```
1130 GOTO 20
1140 '
1150 PRINT:PRINT "COLOR ";A$;F$;
" IS"
1160 IF Y=-1 THEN PRINT "NO ROOM
 FOR":GOSUB 1180:PRINT "OR ENTER
  DIFFERENT PATTERN FOR":RETURN
1170 PRINT "NOT DEFINED":GOSUB 1
180:PRINT "OR ENTER VALID VALUE
FOR":RETURN
1180 PRINT "USE  X  TO STOP":PRI
NT "      N  TO DEFINE NEW ONE":R
ETURN
1190 '
1200 X0=VAL(F$) 'SET COLOR PATTE
RN FOR
1210 IF A$=" " THEN 1280
1220 IF A$="T" THEN 1350
1230 IF A$="M" THEN 1430
1240 IF A$="B" THEN 1500
1250 IF A$="R" THEN 1570
1260 Y=1:RETURN
1270 '
1280 IF X0>8 THEN Y=1:RETURN
1290 X1=PEEK(Q2):IF X1>15 THEN Y
=-1:RETURN
1300 X2=Q+X1*2
1310 POKE X2,X9:POKE X2+1,X0
1320 POKE Q2,X1+1
1330 RETURN
1340 '
1350 IF X0>PEEK(T2+3) THEN Y=1:R
ETURN
1360 X1=PEEK(T2):IF X1>15 THEN Y
=-1:RETURN
1370 X2=T+(X0-1)*5:FOR X0=0 TO 4
1380 POKE T1+X0,PEEK(X2+X0):NEXT
 X0 'MOVE PATTERN
1390 POKE T1,X9:POKE T2,X1+1
1400 T1=T1+5 'MOVE POINTER
1410 RETURN
1420 '
1430 IF X0>PEEK(M2+3) THEN Y=1:R
ETURN
1440 X1=PEEK(M2):IF X1>15 THEN Y
=-1:RETURN
1450 X2=M+(X0-1)*8:FOR X0=0 TO 7
1460 POKE M1+X0,PEEK(X2+X0):NEXT
 X0
1470 POKE M1,X9:POKE M2,X1+1
1480 M1=M1+8:RETURN
1490 '
1500 IF X0>PEEK(B2+3) THEN Y=1:R
ETURN
1510 X1=PEEK(B2):IF X1>15 THEN Y
=-1:RETURN
1520 X2=B+(X0-1)*11:FOR X0=0 TO
10
1530 POKE B1+X0,PEEK(X2+X0):NEXT
 X0
1540 POKE B1,X9:POKE B2,X1+1
1550 B1=B1+11:RETURN
1560 '
1570 IF X0>PEEK(R2+3) THEN Y=1:R
ETURN
1580 X1=PEEK(R2):IF X1>15 THEN Y
=-1:RETURN
1590 X2=R+(X0-1)*8:FOR X0=0 TO 7
1600 POKE R1+X0,PEEK(X2+X0):NEXT
 X0
1610 POKE R1,X9:POKE R2,X1+1
1620 R1=R1+8:RETURN
1630 '
1640 'SETUP POINTERS ETC
1650 BP=&H665B 'SET PRINT LINE
1660 Q=PEEK(&HB7)*256 'START DIR
ECT PAGE
1670 Q1=&H68F6 'LOC PIX BUF WIDT
H
1680 Q2=&H66AD 'LOC NO NORMAL CO
LORS
1690 Q3=&H6590 'EXEC SET PIX TOP
 BORDER
1700 Q4=&H67C0 'EXEC REVERSE PIX
ILS
1710 IX=0 'SWITCH TO REVERSE PIX
ILS
1720 '2 COLORED PATTERNS
1730 T=&H6B14 'ALL PATTERNS IN M
EMORY
1740 T1=&H6914 'PATTERNS USED FO
R DUMP
1750 T2=&H68FE 'LOC NO OF PATTER
NS USED
1760 'ROW COLORS
1770 R=T+5*28 'START IN MEMORY
1780 R1=&H6964 'USED FOR DUMP
1790 R2=&H690D 'LOC OF NUM
1800 'MULTI COLS
1810 M=R+8*25 'IN MEMORY
1820 M1=&H69E4 'USED DUMP
1830 M2=&H6903 'LOC OF NUM
1840 'COL BLOCKS
1850 B=M+8*100 'STASRT IN MEM
1860 B1=&H6A64 'USED IN DUMP
1870 B2=&H6908 'LOC OF NUM
1880 '
1890 BN$(1)="LEFT BORDER":BN$(2)
="RIGHT BORDER":BN$(3)="TOP BORD
ER":BN$(4)="BOTTOM BORDER"
1900 BN$(5)="LEFT MARGIN":BN$(6)
="RIGHT MARGIN":BN$(7)="TOP MARG
IN":BN$(8)="BOTTOM MARGIN"
1910 RETURN
1920 '
1930 PRINT:PRINT "FOR THESE OPTI
ONS":PRINT "PRESS ENTER TO BYPAS
S":PRINT "OR ENTER A KEY TO SELE
CT"
1940 PRINT:INPUT "PARTIAL SCREEN
";D$: IF D$<>"" THEN GOSUB 2010
1950 PRINT:INPUT "NORMAL ROTATIO
N";E$
1960 INPUT "SCALE FACTOR";F$:P2=
VAL(F$):IF P2=0 THEN 1960
1970 IF E$="" THEN IF NA*P2<=WP
THEN RETURN
1980 IF ND*P2 <=WP THEN RETURN
1990 PRINT "SIZE TOO BIG":GOTO 1
930
2000 '
2010 PRINT:PRINT "SCREEN MARGIN
SIZES TO BE ENTER-ED AS NUMBER O
F SCREEN PIXILS"
2020 INPUT "LEFT MARGIN";SM(1)
2030 INPUT "RIGHT";SM(2):INPUT "
TOP";SM(3):INPUT "BOTTOM";SM(4)
2040 FOR X=1 TO 2:IF P1=3 THEN S
M(X)=SM(X)/2
2050 SM(X)=INT(SM(X)/NP)*NP:NEXT
 X
2060 NA=NA-SM(1)-SM(2)
2070 ND=ND-SM(3)-SM(4)
2080 RETURN
2090 '
2100 PRINT:PRINT "ENTER COLOR OF
 BORDER":INPUT "OR X TO SELECT D
IFFERENT";F$
2110 IF F$<>"X" THEN FOR X=1 TO
4:BM$(X)=F$:BN(X)=P3:NEXT X:RETU
RN
2120 PRINT "ENTER SIZE AND COLOR
 OF"
2130 FOR X=1 TO 4:PRINT BN$(X);:
INPUT BN(X),BM$(X):NEXT X
2140 RETURN
2150 '
2160 PRINT:INPUT "ENTER SIZE OF
LEFT MARGIN";BN(5)
2170 BN(6)=WP-BN(1)-BN(2)-BN(5):
IF E$="" THEN BN(6)=BN(6)-NA*P2
ELSE BN(6)=BN(6)-ND*P2
2180 IF BN(6)<0 THEN PRINT "LEFT
 MARGIN TOO LARGE":GOTO 2160
2190 IF BN(5)<0 OR BN(5)>511 OR
BN(6)>511 THEN PRINT "CAN'T DO T
HAT":GOTO 2160
2200 PRINT:PRINT "PRESS ENTER FO
R WHITE MARGINS   OR ENTER COLOR
 OF LEFT MARGIN":INPUT F$
2210 IF F$="" THEN RETURN ELSE B
M$(5)=F$
2220 INPUT "ENTER COLOR RIGHT MA
RGIN";BM$(6)
2230 PRINT "ENTER SIZE & COLOR O
F":FOR X=7 TO 8:PRINT BN$(X);:IN
PUT BN(X),BM$(X):NEXT X
2240 RETURN
2250 '
2260 'MARGINS FOR CENTRE
2270 BN(6)=WP-BN(1)-BN(2):IF E$=
"" THEN BN(6)=BN(6)-NA*P2 ELSE B
N(6)=BN(6)-ND*P2
2280 BN(5)=INT(BN(6)/2):BN(6)=BN
(6)-BN(5)
2290 RETURN
2300 'DEFINE NEW COLOR FOR DUMP
2310 PRINT:PRINT "THIS ALLOWS TE
MPORY DEFINITION  IN EXECUTE TIM
E TABLE ONLY AND  IT MUST BE A T
WO COLORED PATTERN"
2320 Y=0:IF PEEK(T2)>15 THEN PRI
NT "YOUR ONLY OPTION IS A SINGLE
":INPUT "COLOR - ENTER COLOR";X0
:GOTO 1280
2330 INPUT "ENTER TWO COLORS REQ
UIRED";X0,X1
2340 POKE T1,X9:POKE T1+1,0:POKE
 T1+2,0:POKE T1+3,X0:POKE T1+4,X
1
2350 POKE T2,PEEK(T2)+1:T1=T1+5
2360 RETURN
2370 'SET EXTRA MENU
2380 PRINT:PRINT:INPUT "DO MIRRO
R IMAGE (Y/N)";A$
2390 IF A$="Y" THEN IF IX<>0 THE
N IX=0 ELSE IX=1
2400 RETURN
10000 HSCREEN 2:HCLS
10010 HLINE (0,0)-(0,30),PSET
10020 HLINE (0,0)-(30,15),PSET
10030 HLINE (0,30)-(30,15),PSET
10040 HLINE (100,0)-(100,30),PSE
T
```

```
10050 HLINE (100,0)-(130,15),PSE
T
10060 HLINE (100,30)-(130,15),PS
ET
11000 GOTO 11000
```

Listing 5

```
1900 'ROUTINE ADDED TO "FOOTY" T
O EXTRACT EACH BADGE INTO A DIRE
CT ACCESS DISK FILE
1910 'ORIGINAL PROGRAM "FOOTY" B
Y JOY WALLACE, SOFTGOLD, MARCH 1
987
1920 '
1930 'PROGRAM FROM LINE 2500 SHO
ULD BE EXECUTED BY  RUN  2500, B
EFORE RUNNING "FOOTY"
1940 'PLUS THE LOOP ON BREAK KEY
'S IS REPLACED WITH GOSUB 2000
1950 ' AND A CLEAR AND DIM STATE
MENT ADDED TO START
1960 '
2000 UQ=UQ+1:IF UQ=5 OR UQ=9 OR
UQ=13 THEN SCREEN 0:PRINT:INPUT
"ENTER NEXT DISK";VQ:POKE &HE6C6
,18:POKE &HE6C7,18:HSCREEN 2:POK
E &HE6C6,141:POKE &HE6C7,16
2010 FOR X=0 TO 15:V(X)=0:NEXT X
2020 N$="FLAG"+MID$(STR$(UQ),2):
 OPEN "D",#1,N$,160: FIELD #1, 1
60 AS QU$
2030 POKE &HE6E1,&H7E:POKE &HE6E
2,14:POKE &HE6E3,3
2035 POKE &HE00,&H20:POKE &HE01,
0
2040 FOR X=1 TO 192: HCLS: PUT#1
,X
2050 EXEC &HE2A: Y1=&H1DA
2060 FOR Y=0 TO 15
2070 V(Y) = V(Y) + PEEK(Y1+Y) +
PEEK(Y1+Y+16)
2080 NEXT Y
2090 NEXT X
2100 CLOSE: PRINT#-2,N$
2105 POKE &HE6E1,&H8E:POKE &HE6E
2,&H20:POKE &HE6E3,0
2110 FOR X=0 TO 15
2120 PRINT#-2,USING "####";X;PEE
K(&HFFB0 + X)-64;
2130 PRINT#-2,USING "####,###";V
(X)
2140 NEXT X:RETURN
2150 '
2500 M=&HE00
2510 LN=3000:FOR X=0 TO 87 STEP
25:IF X<74 THEN N=25 ELSE N=12
2520 GOSUB 2540:NEXT X
2530 '
2540 PRINT LN;:A=0:FOR Y=0 TO N-
1
2550 READ C$:B=VAL("&H"+C$):A=A+
B:POKE M,B:M=M+1
2560 NEXT Y:READ C$:IF A<> VAL("
&H"+C$) THEN PRINT "ERROR IN LIN
E NO";LN:STOP
2570 LN=LN+10:RETURN
2580 '
3000 DATA 20,0,A0,10,FF,2,D2,10,
CE,2,D0,34,66,33,8C,F0,AE,40,10,
8E,9,89,E6,42,A6,A88
3010 DATA 80,A7,A0,5A,26,F9,AF,4
0,35,66,10,FE,2,D2,7E,E6,EB,8E,1
,DA,C6,28,6F,80,5A,C9B
3020 DATA 26,FB,CE,9,89,E6,C4,4F
,59,49,59,49,59,49,59,49,8E,1,DA
,6C,86,A6,C0,84,F,B56
3030 DATA 8E,1,EA,6C,86,11,83,A,
89,26,E1,39,4D2
3040 END
3041 SAVE"318F:3":END'9
```

Listing 6

```
0 GOTO10
3 SAVE"318G:3":END'9
10 'CALLED SAVPMODE
20 'BASIC PROGRAM TO EXTRACT PMO
DE 3 SCREEN - AND SAVE IT TO A D
IRECT ACCESS DISK FILE
30 'USING VARPTR'S OF STRINGS
40 '
50 OPEN "D",#1,"NAME",32
60 FIELD #1, 32 AS BF$
70 A$="X":A=0:B=0:C=0:X=0
80 A=VARPTR(A$):POKE A,32
90 B=PEEK(&HBA):C=0
100 FOR X=1 TO 192
110 POKE A+2,B:POKE A+3,C
120 LSET BF$=A$: PUT #1,X
130 C=C+32: IF C>=256 THEN B=B+1
:C=0
140 NEXT X
150 CLOSE:STOP
```

■

From p9

```
760 X=5:FOR Y=0 TO 31:SET(X,Y,4)
:NEXT Y
770 X=10:FOR Y=0 TO 31:SET(X,Y,2
):NEXT Y
780 X=15:FOR Y=0 TO 31:SET(X,Y,4
):NEXT Y
790 X=20:FOR Y=0 TO 31:SET(X,Y,2
):NEXT Y
800 X=25:FOR Y=0 TO 31:SET(X,Y,4
):NEXT Y
810 X=30:FOR Y=0 TO 31:SET(X,Y,2
):NEXT Y
820 X=35:FOR Y=0 TO 31:SET(X,Y,4
):NEXT Y
830 X=40:FOR Y=0 TO 31:SET(X,Y,2
):NEXT Y
840 X=45:FOR Y=0 TO 31:SET(X,Y,2
):NEXT Y
850 X=50:FOR Y=0 TO 31:SET(X,Y,4
):NEXT Y
860 X=55:FOR Y=0 TO 31:SET(X,Y,2
):NEXT Y
870 X=60:FOR Y=0 TO 31:SET(X,Y,4
):NEXT Y
880 RETURN
```

Hints and Tips

Where's that error again?

Okay, so there's this error in my Basic program. And fair enough I did a bit of variable experimentation to find this error. And now I can't remember that line number, because I've used more than a screenload of typing ... aaarghh!

(Luckily this programmer can now flip through CoCo Magazine and find this little program. It will return the last program line executed.)

```
PRINTPEEK(43)*256+PEEK(44)
```

HSCREEN bugs (CoCo 3)

Whenever you execute a HSCREEN command, a HCLS command is executed automatically without you knowing/ wanting it.

Here is a way to defeat that.

```
POKE &HE6C6,18:POKE &HE6C7,18
```

Booting OS9 on a DECB 1.0/2.0 DOS

Just type the below program and you'll be able to boot any bootable OS9 disk from your DECB 1.0/2.0 DOS.

(For people who can't use the DOS command available in DECB 1.1/2.1:

```
10 FOR I=0 TO 70:READ A$:POKE &H
5000+I,VAL("&H"+A$) 40 NEXT I:CL
S:PRINT"INSERT  OS9  DISKETTE"
15  PRINT"INTO DRIVE 0 AND PRESS
A KEY"
70          EXEC44539:EXEC&H5000
20   DATA   86,22,8E,26,00,8D,0D
25   DATA   FC,26,00,10,83,4F,53
30   DATA   26,03,7E,26,02,39,34
35   DATA   20,10,BE,C0,06,A7,22
40   DATA   86,02,A7,A4,6F,21,6F
45   DATA   23,6C,23,AF,24,10,BE
50   DATA   C0,06,A6,23,81,13,27
55   DATA   12,AD,9F,C0,04,4D,27
60   DATA   06,6C,23,6C,24,20,E9
65   DATA   7F,FF,40,35,A0,4F,20
70 DATA F8
```

CONTENTS

By Nicholas Fuller

UTILITY
COCO 3

ONE OF THE disadvantages of a tape based system is unlike a disk drive you can't get a directory of what is stored on the cassette. This is inconvenient as one often forgets what is on the tape.

"Contents" for CoCo 3 is a directory which allows you to enter the names of the programs you have on your tape for quick reference, which can be edited.

When RUN, you will be asked to enter your security code (input statement, line 140). You can change it if you wish.

Apart from that, it's pretty simple - I have placed garbage in the data lines just to fill the pages. You can replace this with your own programs.

```
0 GOTO10
1 '********** CONTENTS *********
   **** NICHOLAS FULLER ********
3 SAVE"316A:3":END'8
10 ON BRK GOTO 180:HSCREEN 2:PAL
ETTE 0,0:HCIRCLE(40,30),15,1,1,.
25,.75
20 HCIRCLE(40,30),10,1,1,.25,.75
30 HDRAW"BM40,40;C1;D3;B;U20;U6"
:HPAINT(34,22),1,1
40 HCIRCLE(60,30),15,1:HCIRCLE(6
0,30),10,1:HDRAW"BM85,17;C1;L7;D
28;R7;U15;F15;R7;U28;L7;D15;H15"
:HPAINT(53,22),3,1:HPAINT(87,20)
,6,150 HDRAW"BM113,17;C1;D7;R7;D
21;R6;U21;R7;U7;L20":HPAINT(115,
20),7,1
60 HDRAW"BM138,17;C1;D28;R12;U6;
L5;U5;R5;U6;L5;U5;R6;U6;L12":HPA
INT(140,20),5,1
70 HDRAW"BM159,17;C1;D28;R7;U15;
F15;R7;U28;L7;D15;H15;L7":HPAINT
(162,18),4,1
80 HDRAW"BM195,17;C11;D7;R7;D21;
R6;U21;R7;U7;L20":HPAINT(197,20)
,2,11
90 HCIRCLE(230,25),10,1,1,.4,.87
5:HCIRCLE(230,24),4,1,1,.325,.05
:HCIRCLE(226,38),10,1,1,.8,.4:HC
IRCLE(226,39),4,1,1,.85,.5:HPAIN
T(227,22),9,1
100 HDRAW"BM200,200;C1;L100":HLI
NE(235,23)-(238,20),PSET:HLINE(2
21,38)-(219,43),PSET:HLINE(223,3
0)-(227,37),PSET:HLINE(228,26)-(
232,31),PSET:HPAINT(229,28),9,1
110 HCOLOR 5,8:HPRINT(7,20),"ENT
ER YOUR SECURITY CODE":HPRINT(10
,10),"BY NICHOLAS FULLER"
120 HLINE(150,150)-(120,120),PSE
T,BF
130 SOUND 12,3:I$=INKEY$:IF I$="
" THEN 130 ELSE 140
140 INPUTA$:IF A$="CO CO"THEN 15
0 ELSE 180
150 LPOKE516158,34:WIDTH 40:LPOK
E516158,56:ATTR 3,2:PRINT"WELCOM
E TO *";:ATTR4,5:PRINT"  C o n t
 e n t s":ATTR 6,3:PRINT"CONTENT
S IS A PROGRAM WHICH ALLOWS YOU
TO ENTER THE NAMES OF THE PROGRA
MS ON YOUR TAPE AND USE IT AS AN
 INDEX"
160 PRINT"THE INDEX MAY BE UPDAT
ED AND ALLOWS YOU TO SHOW THE NA
ME/DESCRIPTION/COUNTER NUMBER OF
 YOUR PROGRAM":PRINT" PRESS <C>
KEY":INPUTA$:IF A$="C" THEN 190
ELSE 170
170 GOTO 180
180 WIDTH40:ATTR3,2,B:LPOKE51615
8,45:LPOKE516158,47:PRINT:PRINT:
PRINT:PRINT"NEGATIVE ADMITTANCE-
------SECURITY SHELL NOT PENETRA
TED":FORT=255TO1 STEP -1:SOUND T
,1:NEXT:GOTO 180
190 CLS:PRINT"MENU":PRINT"1-EDIT
 PAGE 1 /2":PRINT"2-ESCAPE":PRIN
T"3-VIEW PAGE 1":PRINT"4-VIEW PA
GE 2":PRINT "5-LOAD A PROGRAM":I
NPUT A:ON A GOSUB 240,210,260,34
0,500
200 WIDTH 40:CLS:PLAY"T8;V10;O4;
EEDACBF#GDDAACA"
210 CLS:INPUT"ARE YOU SURE YOU W
ISH TO STOP Y/N";A$:IF A$="N"  T
HEN 190:IF A$="Y"THENSTOP
240 CLS:PRINT"INPUT NUMBER TO BE
 EDITED ON PAGE 1":INPUTA: ON A
GOSUB 1000,1001,1002,1003,1004,1
005,1006,1007,1008,1009,1010,101
1,1012,1013,1014,1015,1016,1017,
1018,1019,1020,1021,1022,1023,10
24,1025,1026,1027,1028,1029,1030
241 PRINT"ENTER NUMBER TO BE EDI
TED ":INPUTA:ON A EDIT 360,361,3
70
260 CLS:ATTR4,5,U:RESTORE:A=7:PR
INT"***C O N T E N T S***":ATTR4
,7:PRINT:PRINT:PRINT"Tape number
:";A
270 ATTR 2,1,B:PRINT"NAME
  TYPE      COUNTER NO.":ATTR 4,
7
271 GOTO 310
272 STOP
278 EDIT 280
280 DATA 1, LUNAROVER,GAME,27
281 DATA 2,POO',GAME,23
282 DATA 3,HHH,WWW,22
283 DATA 4,GAME,5,6
284 DATA 6,ED,ED,ED
285 DATA SD,DF,ED,FD
286 DATA 7,FF,GG,RR
287 DATA 8,WW,FF,YY
288 DATA Q,S,DD,F
289 DATA 10,3,3,3
290 DATA 2,22,22,22
291 DATA 23,43,655,34
292 DATA D ,E,EW,W
293 DATA 23,RF,ED,S
294 DATA 23E44,EE,EE,EE
295 DATA POOYAN22 2,BB,V,C
310 FORY=1TO15:READA$,B$,C$,D$:W
=W+1:LOCATE1,6+Y:PRINTA$;:LOCATE
3,6+Y:PRINTB$:LOCATE16,6+Y:PRINT
 C$:LOCATE30,6+Y:PRINTD$:NEXT
319 PRINT W
320 INPUT"PRESS <C> TO RETURN TO
 MENU";Q$:IF Q$="C"THEN GOTO190
330 STOP
340 CLS:ATTR6,0:A=7:PRINT"***C O
 N T E N T S***":ATTR4,3:PRINT:P
RINT:PRINT"Tape number";A
348 GOTO 350
349 RESTORE E$,F$,G$,H$
350 ATTR 6,5,B:PRINT"NAME
   TYPE      COUNTER NO."
351 DATA R,R,R,R
352 DATA S,S,S,S
353 DATA T,T,T,T
354 DATA U,U,U,U
355 DATA C,C,C,C
356 DATA E,E,E,E
357 DATA Q,Q,Q,Q
358 DATA X,X,X,X
359 DATA E,E,E,E
360 DATA M,M,M,M
361 DATA D,D,D,D
362 DATA W,W,W,W
363 DATAQ,W,FD,GGGGF
364 DATA D,D,D,D
365 '
366 '
380 FORY=1TO15:READE$,F$,G$,H$:W
```

Continued on p58

OS9 LEVEL 2 Screen Patches

By Rob Unsworth

THOSE OF YOU WHO have monochrome monitors will no doubt be aware of the problem of using the 80 column display with the various windows that are on your OS9 Level 2 system disk. In most cases, due to the colour combinations, it is almost impossible to read.

Then there is the problem that after having configured a disk using TERM.WIN your screen boots up 40 column with black lettering on a green background.

Having an amber monitor I found that the only foreground-background colour combinations which are acceptible is white and black.

Why not have an 80 column display that is not directly accessible? ... it is!

Using modpatch you can rebuild any screen to your own specifications.

Firstly you'll need to decide what size window you wish to create. On startup it is $28 (40 columns) and $18 (24 lines).

The offset for these bytes are at $2C and $2D. To create an 80 column screen, the offset for the screen type will also need to be changed.

This is at offset $30, the startup byte is $01 (40 column) and needs to be changed to $02 (80 column).

The offset for the no of lines will not have to be changed as it is already set for 24 lines.

If for example 12 lines were wanted then you would change $2d from $28 to $1c.

The offsets for the screen colours are $33 (text), $34 (background) and $35 (border).

On startup these are $02, $03 and $03 respectively.

Now to get that clear readable 80 column screen with white text on a black background, type the following :-

```
modpatch 1 term  c 2c 28 50 c 30 01 02 c 33 02 00 c 34 03 02 c 35 03 02 v
```

Then all you need to do is a "cobbler /d0" to generate a bootable system disk, with /term as a 80 column white on black screen.

If you want to have the pause command built in then change the byte at offset $19 from $00 to $01.

To change windows w1 to w7, to either full screen or multiple windows on the same screen, then the bytes at offsets $31 and $32 will need to be changed, as they determine the x-y position of the screen.

To get these and the other bytes that need patching.

Copy the dump command from level 1, then dump the device descriptors from the modules directory on the config disk.

From p57

```
=W+1:LOCATE1,6+Y:PRINT E$;:LOCAT
E 3,6+Y:PRINTF$:LOCATE16,6+Y:PRI
NT G$:LOCATE30,6+Y:PRINT H$:NEXT
381 PRINT W
382 RESTORE
383 ATTR3,3
390 INPUT"PRESS C TO RETURN TO M
ENU";N$:IF N$="C"THEN190
500 CLS:INPUT"ENTER NAME OF PROG
RAM";Z$:INPUT"WHAT IS THE MODE 1
e 1 basic 2 machine code 3 OS9 "
;B$:IF B$="1"THEN510:IF B$="2"TH
EN520:IF B$="3"THEN 530
510 PRINT"PLACE TAPE AT REVELLAN
T TAPE NO.WITH PLAY BUTTON DOWN
HIT C WHEN READY":INPUTC$:IF C$=
"C"THEN 511
511 CLOAD Z$
520 PRINT"PLACE TAPE AT REVELLAN
T NO. WITH PLAY BUTTON DOWN":INP
UT"DOES IT REQUIRE AN OFFSET ADD
RESS Y/N";D$:IF D$="N" THEN CLOA
DM J$:IF D$="Y"THEN521
521 INPUT"ENTER ADDRESS";E$:PRIN
T"PLACE TAPE AT REVALANT TAPE NO
. WITH PLAY BUTTON DOWN HIT C WH
EN READY":INPUTC$:IF C$="C"THEN
CLOADM J$,E$
530 CLS:PRINT"USE OS9 METHOD!":G
OTO 530
1000 EDIT 280
1001 EDIT 281
1002 EDIT 282
1003 EDIT 283
1004 EDIT 284
1005 EDIT 285
1006 EDIT 286
1007 EDIT 287
1008 EDIT 288
1009 EDIT 289
1010 EDIT 290
1011 EDIT 291
1012 EDIT 292
1013 EDIT 293
1014 EDIT 294
1015 EDIT 295
1016 EDIT 351
1017 EDIT 352
1018 EDIT 353
1019 EDIT 354
1020 EDIT 355
1021 EDIT 356
1022 EDIT 357
1023 EDIT 358
1024 EDIT 359
1025 EDIT 360
1026 EDIT 361
1027 EDIT 362
1028 EDIT 363
1029 EDIT 364
1030 EDIT 365
```

LOST IN THE WILDS?

by Ozzie OS9

(from the depths of PNG)

SOME OF YOU HAVE probably been wondering whats happened to Fred? Haven't heard him carrying on about OS9 lately! (One or two were probably pleased)

Well I have bad news ... I'm BACK! Well, not in person anyway. In May I packed my bags and headed for Papua New Guinea.

It was rather short notice, so all you people who have been trying to phone me, or expected replies to letters or Viatel MB's:- all I can say is SORRY.

Unfortunately Australia Post don't upgrade local mail to overseas air mail without getting paid hard cash up front. I have now replied to all letters and you should have had them for some time (by the time you read this in the mag).

For those of you who wish to phone (expensive) my number is:
(ISD) 0011 (country) 675 (number) 255565 OR 255011.

Address is:-
P.O. Box 5447,
BORDKO,
Papua New Guinea.

Boroko is a suburb of Port Moresby, and aerograms are a lot cheaper than phoning. I will 'ATTEMPT' to get onto Viatel from time to time. It is rather difficult due to the amount of line noise and echo which causes all sorts of problems for the transfer of data, so please be patient for a reply - better still just write to the above address.

Recieved copies of August and September's mags yesterday afternoon and having devoured the contents of them twice, I would like to congratulate the winners of prizes and awards at CoCoConf, as well as people like Johanna Vagg (Hi Johanna), Mike Turk and many others who freely give their time to make CoCoConf a success.

I was dissapointed to hear that Sydney siders did not support the Conf. What a shame - they don't know what they have missed. I'm only sorry that I was not able to be there to meet up with the many friends I had met at previous Conf's, as well as the opportunity to meet many more.

As I haven't seen a CoCo or Softgold since May, going through the newly arrived August/Sept. mags, I am suddenly reminded of the continually high standard of articles and programs submitted.

These are equal to, and in many cases better than anything produced for the the home computer anywhere in the world. Personal opinion you say!

Okay, that may be true, but ask people like the Canadians who have a multitude of magazines to choose from (local, USA & others) - you may be surprised at their answer.

So all the novices to the CoCo, do not be discouraged, the articles and programs in the magazine in many cases are submitted by people who not so long ago were new to the CoCo too.

If you do not belong to a user group, join now, or even start your own. User groups are a very good way of giving and recieving help and information.

Since my arrival in PNG my access to information has been resticted by the number of known CoCo owners in Port Moresby (one - me) and by my job.

So in order to be able to collect enough information for regular articles I'm going to have to rely heavily on your letters and questions. That way we help each other, as well as others new to the magazine and CoCo. Although my major interest lies in OS9, queries on any matter are always welcome.

So thats it from the WILD'S of PAPUA NEW GUINEA.

November's CoCoOz

Next month's Australian CoCo Magazine will feature a number of OS9 programs and CoCoOz ON DISK will have these programs on it!

The disk will be a "flippy" – RS DOS programs from the magazine on one side, and OS9 programs on the other.

Dependent upon your response to this new system, we will look at making CoCoOz this way more often.

Tape users please note that you will only get the programs found on the RS DOS side of the CoCoOz disk.

SUBMITTING YOUR WORK

Ah! So you've finally finished that program? And you say to yourself, "What a great program that would make for CoCo Magazine/Softgold Magazine!"

And so you wonder to yourself, "How am I going to send this program in to the magazine?". Some time goes by and you suddenly realise, "Hey, there's an article in this month's magazine about submitting your work. I'll read through that and maybe that'll help me."

So you rip the magazine out of your stack of other CoCo/Softgold magazines and read the article on how to submit your program.

It reads ...

" ... we accept programs stored on both tape and disk ONLY along with a hard copy of the program(s) (optional only; we use it here as a reference to see what the program is/does) and suitable instructions.

Saving to Tape

Each program would be best saved three times with the last save being in ASCII. The tapes we recommend you use are either a C30 or less (the reason for that is that tapes longer than C30 have a tendancy to tear).

It'd be even better if you could include some instructions along with the program, either as a seperate program or in the wordprocessors listed below.

Saving to Disk

With disk, you'd be best to save it three times with the last save being in ASCII. Also, the extension name for the second and third copy should be different, so to distinguish the three copies. A simulation is given below.

' ... I have just saved 3 copies of a program called "HORSE". The directory listing would be:

```
HORSE  BAS  0 B 3
HORSE   1   0 B 3
HORSE   2   0 A 3'
```

Any instructions could be saved in the same system using either a program or in the wordprocessors listed below.

Wordprocessors we use.

Here is a list from our most preferable worprocessors to the drastic measure one could take to tell us how your program works.

1. Telewriter/Telepatch
2. Scripsit
3. PenPal
4. VIP Writer
5. Any form of data file.
6. Instructions written in a seperate program."

"Oh wow!", you think to yourself as you read it with awe and astonishment. So you go about your busy little way saving your program and instructions to tape or disk. Then you say to yourself, "Where do I send it?"

You read the article on ...

" ... any articles and programs should be sent to this address:

Submissions Editor,
Freepost 5
PO Box 1742,
Southport, Qld, 4215

All mail to this address need not be paid for.

All tapes and disks received will be returned after three months in case we need to refer to something or re-print something."

So place your tape/disk along with your hardcopy of the listing in a postpack (or suitable wrapping) and pop it in the mail.

All done!!

Telecom VIATEL

AUSTRALIA'S NATIONAL VIDEOTEX SERVICE

APPLICATION FORM

DATE OF APPLICATION / /

(BEFORE COMPLETING THIS APPLICATION, PLEASE READ REVERSE SIDE CAREFULLY)

section 1

PLEASE TICK APPROPRIATE BOX TO INDICATE SERVICE REQUIRED

BUSINESS SERVICE ☐ NON-BUSINESS SERVICE ☐

(CHARGES INCURRED ON BUSINESS SERVICES ARE USUALLY TAX DEDUCTIBLE)

SURNAME (OR BUSINESS NAME IF BUSINESS SERVICE) GIVEN NAMES

POSTAL ADDRESS NUMBER/STREET

SUBURB/CITY STATE POSTCODE

TELEPHONE NUMBER ON WHICH SERVICE IS REQUIRED (INCLUDING STD CODE)

section 2

CONTACT NAME (IF BUSINESS SERVICE) GIVEN NAMES

POSTAL ADDRESS FOR BILLING IF DIFFERENT FROM SECTION 1 ABOVE

NUMBER/STREET

SUBURB/CITY STATE POSTCODE

CONTACT TELEPHONE NUMBER (INCLUDING STD CODE)

section 3

PLEASE DESCRIBE NATURE OF BUSINESS (OR OCCUPATION IF NOT A BUSINESS SERVICE)

PLEASE INDICATE TYPE OF EQUIPMENT USED TO ACCESS VIATEL

special instructions

THIS FORM SHOULD BE HANDED IN AT ANY TELECOM BUSINESS OFFICE OR MAY BE MAILED WITHOUT A STAMP TO FREEPOST 20, VIATEL BOX 188C, GPO MELBOURNE, VICTORIA 3001

PLEASE ALLOW TEN WORKING DAYS FOR PROCESSING OF APPLICATION AND RETURN MAIL ADVICE.

telecom use only

DTE PP VN

BG SC CI

REF

GOLDLINK: The place to be on VIATEL

SPAN OVERLAY NO. 140

REGISTRATION AND SUBSCRIPTIONS

Customers must register as a Business Service if the telephone number nominated for the use of the VIATEL Service is a Business Service and/or VIATEL is to be used wholly or mainly for Business, Commercial, Industrial, Professional or Government purposes. (Charges incurred on Business Services are usually tax deductible.)

Where a Business Telephone Service is nominated for the use of VIATEL, but the use of VIATEL is wholly or mainly for Non-Business purposes, the Customer may be registered as a Non-Business VIATEL subscriber, providing the registration is taken out in the Customer's personal name and address and not a Business name.

Telecom Australia will register the Business or Individual named under Section 1 as a Customer of its VIATEL Service and will provide the Customer with a confidential Customer Identity Number and Personal Password by mail.

Where billing address is indicated, bills and bill related correspondence ONLY will be forwarded to that address. All other correspondence will be forwarded to address under Section 1.

Customers should advise VIATEL of any change of address as soon as possible.

If you lose your Customer Identity Number and/or Personal Password, you must advise VIATEL in writing before new numbers are issued. Our postal address is: Freepost 20, Box 188C, GPO Melbourne, Vic. 3001. FOR SECURITY REASONS REPLACEMENT NUMBERS AND PASSWORDS CANNOT BE PROVIDED OVER THE TELEPHONE.

Customers of VIATEL acknowledge that their name and registered VIATEL Number will appear on the VIATEL Mailbox Directory and that Service Providers and/or other registered VIATEL users may send messages to their VIATEL number.

Telecom Australia undertakes no responsibility in relation to the accuracy of the information or service provided by Service Providers on VIATEL. Telecom Australia will not be responsible for any loss or damage arising out of or in any way connected with the use of this information or service.

Attention is also drawn to the terms and conditions governing the provision of information and services by some Service Providers. These terms and conditions may, in some cases, include a disclaimer absolving the Service Provider from liability regarding information and services supplied on VIATEL. The means of accessing these terms and conditions is set out on the Service Provider's Index Page on VIATEL.

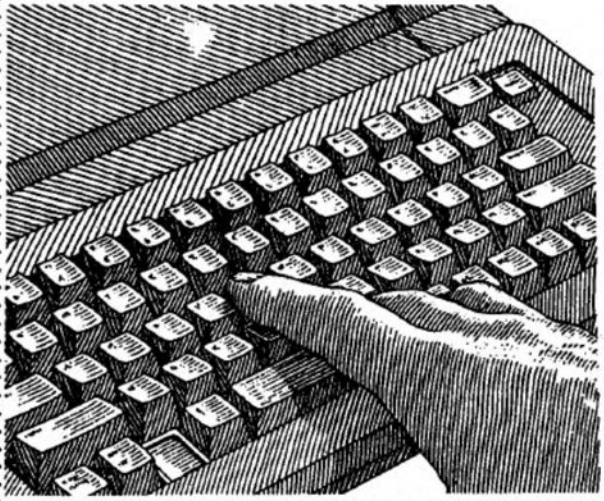
Should you require any changes to your existing telephone equipment (e.g. new exchange line, additional socket), please contact your local District Telecom Office.

In a small number of cases VIATEL reception may be unsatisfactory. Correction may incur an additional charge.

COMPETITIONS

As a result of the success of the Tandy programming contest this year, Tandy have agreed to rerun it in 1987–88!

So – get your thinking caps on! Perhaps YOU will be the one receiving that cheque from Tandy next year!



And speaking of cheques, the best ML Game for the CoCo 3 with a BiCentennial theme submitted by 7th November 1987, will win a $300 prize WITH royalties for every program sold from Goldsoft.

The next minor competition – the annual Graphics Competition begins now and ends on 7th November, 1987.

All computer created pictures are elligible, and the competition is divided into a section for Basic pictures, one for CoCoMax & ColourMax pictures, and one for pictures created in some other way.

As with the last Graphics Competition, the judges are looking for animated pictures.

First prize in each category will be 5 boxes of disks or tapes.

GOLDSOFT

P.O. BOX 1742, SOUTHPORT. QLD. 4215 Phone (075) 39–6177

Goldsoft Price list as at September, 1987 Please tick (✓) your requirements.

HARDWARE

CoCoConnection: $219.95 ( )
Video Amp: With sound - $35.00 ( )
Without sound - $25.00 ( )
The Probe: $49.95 ( )

GOLDLINK
Access Goldlink *642# on Viatel with a 1200/75 baud modem. Annual subscription: $44.95 ( )

SOFTWARE
Magazines, Tapes & Disks

Australian oo (Advanced Programs for your CoCo):

Magazines:	Tape ( )	Disk ( )
12 Months	$39.95 ( )	$123.75 ( )
6 Months	$24.95 ( )	$ 74.25 ( )
1 Month	$ 4.50 ( )	$ 16.50 ( )

Softgold (Programs for your CoCo):

Magazines	Tape ( )	Disk ( )
12 Months	$39.95 ( )	$123.75 ( )
6 Months	$24.95 ( )	$ 74.25 ( )
1 Month	$ 4.50 ( )	$ 16.50 ( )

Golddisk - Available Quarterly:
#1 $16.00 ( )
#2 soon!
#3 not so soon!

The CoCo Tape/Disk:
1 - Tape: $10.00 ( ) Disk: $16.00 ( )
2 - Tape: $10.00 ( ) Disk: $16.00 ( )
3 - Tape: $10.00 ( ) Disk: $16.00 ( )
4 - Tape: $10.00 ( ) Disk: $16.00 ( )

"Say the Wordz":
Two Curriculum based speller programs for your Tandy Speech/Sound pack: $29.95
Req: 32K + Tandy Speech Pack ( )

Best of CoCoOz - $16.00
A selection of programs from Australian CoCo magazine.

	Tape	Disk
# 1 - Education:	( )	( )
# 2 - Games 16K:	( )	( )
# 3 - Games 32K:	( )	( )
# 4 - Utilities:	( )	( )
# 5 - Adventure:	( )	( )
# 6 - Preschool:	( )	( )
# 7 - Graphics:	( )	( )
# 8 - Games 16K:	( )	( )
# 9 - Games 32K:	( )	( )
#10 - Education:	( )	( )
#11 - Education:	-	( )

BRIC-A-BRAC

Blank Tapes:	12 @	$18.00 ( )
(C-30)	1 @	$ 2.00 ( )
Tape cases:	12 @	$ 5.00 ( )
Disk DSDD:	10 @	$20.00 ( )
	1 @	$ 2.50 ( )

BOOKS
Help (for your CoCo): $ 9.95 ( )
Mico Help (for your MC-10): $ 9.95 ( )

BACK ISSUES
Australian CoCo: Sep 84 - Dec 85: $2.00 ( )
Australian CoCo: Jan 86 - Feb 87: $3.75 ( )
Australian Mico: Aug 84 - Dec 85: $2.00 ( )

ADDITIONAL REQUIREMENTS

.....

.....

.....

New Subscription: ( ) Renewal ( )
Sub No:.....

Name:.....

Address:.....

.....

.....P/Code.....

Phone (...).....

Please find enclosed:
a. Cheque: ( )
b. Money Order: ( )
c. Credit card: ( )

Credit Card Type & Number:

Bankcard ( ):.....-.....-.....-.....

Mastercard ( ):.....-.....-.....-.....

Visa ( ):.....-.....-.....-.....

Expiry Date:...../19...

Authorised Amount: $.....

Signed:.....

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

user CONTACTS

ACT:
- CANBERRA NTH — JOHN BURGER 062 58 3924
- CANBERRA STH — LES THURTON 062 86 9226

NSW:

SYDNEY:

BANKSTOWN	PAT DORSETT	02 646 3619
BLACKTOWN	KEITH GALLAGHER	02-627-4627
CARLINGFORD	ROSCO MCKAY	02 624 3353
CHATSWOOD	BILL O'DONNELL	02 419 6081
COLYTON	HERMAN FREDRIKSSON	02 623 6379
FAIRFIELD	ARIH PITTARD	02 72 2881
GLADESVILLE	MARK ROTHWELL	02 817 4627
HILLS DIST	ARTHUR SLADE	02 674 5620
HORNSBY	ATHALIE SMART	02 848 8830
INGLEBURN	STEPHEN RIDGEWAY	02 605 7382
KENTHURST	TOM STUART	02 654 2178
LEICHHARDT	STEVEN CHICOS	02 560 6207
or	GORGE ECHEGARAY	02 560 9665
LIVERPOOL	LEONIE DUGGAN	02-607-3791
MACQUARIE FIELDS	BARRY DARWION	02 618 1909
MINTO	GRAHAM POLLOCK	02 603 5028
SUTHERLAND	IAN ANNABEL	02 528 3391
SYDNEY EAST	JACKY COCKINOS	02 344 9111

ALBURY	RON DUNCAN	060 43 1031
ARMIDALE	DOUG BARBER	067 72 7647
BLAXLAND	BRUCE SULLIVAN	047 39 3903
BROKEN HILL	TERRY MOONAN	080 88 2382
CAMDEN	SEAN MURDOCH	047 74 8291
COFFS HARBOUR	BOB KENNY	066 51 2205
COOMA	ROSS PRATT	064 52 3065
COORANBONG	GEORGE SAVAGE	047 77 1054
COOTAMUNDRA	CHERYLE WILLIS	069 42 2264
DENILIQUIN	WAYNE PATTERSON	058 81 3014
DUBBO	GRAEME CLARKE	068 89 6549
FORBES	JOHANNA VAGG	068 52 2943
GOSFORD	PETER SEIFERT	043 32 7874
GRAFTON	PETER LINDSAY	066 42 2503
GUYRA	MICHAEL J. HARTMANN	067 79 7547
JUNEE	PAUL MALONEY	069 24 1860
KEMPSEY	RICK FULLER	065-62-7222
LEETON	BRETT WALLACE	069-53-2081
LISMORE	BOB HILLARD	066 24 3089
LITHGOW	DAVID BERGER	063 52 2282
MAITLAND	BILL SNOW	049 66 2557
MOREE	ALF BATE	067 52 2405
NAMBUCCA HDS	WENDY PETERSON	065 68 6723
NARROMINE	GRAEME CLARKE	068 89 6549
NEWCASTLE	LYN DAWSON	049 49 8144
NOWRA	ROY LOPEZ	044 46 5449
PARKES	DAVID SMALL	068 62 2682
PORT MACQUARIE	RON LALOR	065 62 2682
SPRINGWOOD	JIM HOFFITT	047 54 1474
TAHMOOR	GARY SYLVESTER	046 81 9318
UPPER HUNTER	TERRY GRAVOLIN	065 45 1698
URALLA	FRANK MUDFORD	067 78 4391
WAGGA WAGGA	CES JENKINSON	069 25 2263
WYONG	JOHN WALLACE	043 90 0312

NT:
- DARWIN — BRENTON PRIOR 089.81.7766

QLD:

BRISBANE:

BIRKDALE	COLIN NORTH	07 824 2126
CLAYFIELD	JACK FRICKER	07 262 8869
COLL'WOOD PK	ANDREW SIMPSON	07 286 5206
IPSWICH	NICK MURPHY	07 271 1777
PINE RIVERS	BARRY CLARKE	07 204 2806
SOUTH WEST	BOB DEVRIES	07 372 7816
SCARBOROUGH	PETER MAY	07 203 6723
WOODRIDGE	ALLAN ALLSOP	07 349 1831
AIRLIE BEACH	GLEN EVANS	079 46 1264

BIGGENDEN	ALAN MENHAM	071 27 1272
BOWEN	TERRY COTTON C/O	077 86 2220
BUNDABERG	RON SIMPKIN	071 71 5301
CAIRNS	JEFF LARSEN	070 54 7127
DALBY	MERRICK TANSKY	074 62 3228
GLADSTONE	CAROL CATHCART	079 78 3594
GOLD COAST	GRAHAM MORPHETT	075 51 0577
GYMPIE	BERT LLOYD	071 8219100
HERVEY BAY	LESLEY HORWOOD	071 22 4989
MACKAY	LEN MALONEY	079511333x782
MARYBOROUGH	JOHN EFFER	071 21 6638
MT ISA	JACK RAE	077 43 3486
MURGON	PETER ANGEL	071 68 1628
ROCKHAMPTON	KIERAN SIMPSON	079 28 6162
TARA	DEBBIE DORFIELD	074 65 3177
TOOWOOMBA	LEN GERSEKOWSKI	076 35 8264
TOWNSVILLE	JOHN O'CALLAGHAN	077 73 2064

SA:

ADELAIDE	JOHN HAINES	08 278 3560
PORT NOARLUNGA	ROB DALZELL	08 386 1647
SEACOMBE HTS	GLENN DAVIS	08 296 7477
PORT LINCOLN	BILL BOARDMAN	086 82 2385
PORT PIRIE	VIC KNAUERHASE	086 32 1230
WHYALLA	MALCOLM PATRICK	086 45 7637

TAS:

DEVONPORT	JEFF BEST	004 24 6759
HOBART	BOB DELBOURGO	002 25 3896
KINGSTON	WIM DE FUIT	002 29 4950
LAUNCESTON	BILL BOWER	003 44 1584
SMITHTON	HARRY CHRISAFIS	004-52-1590
WYNYARD	ANDREW WYLLIE	004 35 1839

VIC:

MELBOURNE:

MELBOURNE CCC	JOY WALLACE	03 277 5182
DANDENONG	DAVID HORROCKS	03 707 5870
DONCASTER	JUSTIN LIPTON	03 857 5149
FRANKSTON	BOB HATTER	03.783.9748
HARRE WARREN	LEIGH EAMES	03 704 6660
NTH EASTERN	PETER WOOD	03 435 2018
N'TON PENNINSULA	GORDON CHASE	059 71 1553
MELTON	MARIO GERADA	03 743 1323
PAKENHAM	JASON HALL	059 41 1398
RINGWOOD	IVOR DAVIES	03 758 4496
SUNBURY	JACK SMIT	03.744.1355
SUNSHINE	IAN BUTTRISS	03 314 3240
UPR F'TREE GLY	RORY DOYLE	03 758 2671
BAIRNSDALE	COLIN LEHMANN	051 57 1545
BALLARAT	MARK BRYBLANDER	053 32 6733
DAYLESFORD	DANNY HEDJI	054 24 8329
GEELONG	DAVID COLLEN	052 43 2128
MAFFRA	MAX HUCKERBY	051 45 4315
MOE	JOSEPH HESTER	051 27 7817
MORNINGTON	MICHAEL MONCK	03 789 7997
MURWELL	JEFF SHEEN	051 33 9904
SHEPPARTON	ROSS FARRAR	058 25 1007
SMYTHESDALE	TONY PATTERSON	053 42 8615
SWAN HILL	HARRIE GERRARD	050.32.2838
TONGALA	TONY HILLIS	058 59 2251
TRARALGON	LEIGH DAWES	051 74 5552
WONTHAGGI	LOIS O'MEARA	056 72 1593

WA:

PERTH	IAIN MACLEOD	09 448 2136
GIRRAWHEEN	HANK WILLEMSEN	09 342 7639
KALGOORLIE	TERRY BURNETT	090.21.5212

CANADA - CoCo:

Ontario	Richard Hobson 416 293 2346
Toronto	Franz Lichtenberg 416 845 2889

special interest groups

TEACHERS' INTEREST GROUP
- BRISBANE — BOB HORNE 07 281 8151

BUSINESS:
- BRIZBIZ — BRIAN BERE-STRETTER 07 349 4696

OS9 GROUPS:
- NATIONAL OS9 USERS' GROUP — GRAHAM NICHOLS 02 451 2954

NSW
- SYDNEY
  - BANKSTOWN — CARL STERN 02 646 3619
  - CARLINGFORD — ROSKO MCKAY 02 624 3353
  - GLADESVILLE — MARK ROTHWELL 02 817 4627
  - SYDNEY EAST — JACKY COCKINOS 02.344.9111
- COOMA — ROSS PRATT 064 52 3065

QLD
- BRISBANE — JACK FRICKER 07 262 8869

VIC
- LATROBE VLY — GEORGE FRANCIS 051 34 5175

WA
- KALGOORLIE — TERRY BURNETT 090.21.5212

MC-10 CONTACTS:
- LISMORE — BOB HILLARD 066 24 3089
- SYDNEY — GRAHAM POLLOCK 02 603 5028

TANDY 1000 / MS DOS:

NSW:
- GLADESVILLE — MARK ROTHWELL 02 817 4627
- SYDNEY WEST — ROGER RUTHEN 047.39.3903
- WYONG — JOHN WALLACE 043 90 0312

QLD:
- PRISBANE
  - NORTH — BRIAN DOUGAN 07 30 2072
  - SOUTH — BARRY CAWLEY 07 390 7946
- GOLD COAST — GRAHAM MORPHETT 075 51 0577

SA
- PORT LINCOLN — BILL BOARDMAN 086 82 2385

VIC:
- LA TROBE VALLEY — PETER FOLEY 051 74 5791
- MELBOURNE — TONY LLOYD 03 882 4664

FORTH:
- SYDNEY — JOHN REDMOND 02 85 3751

ROBOTICS:
- GOLD COAST — GRAHAM MORPHETT 075 51 0577
- SYDNEY — GEOFF FIALA 02 84 3172

CHRISTIAN USERS' GROUP:
- COLLIE — RAYMOND L. ISAAC 097 34 1578

MSX
- FRANKSTON — ALAN HASSELL 03 786 6290

300 BAUD BULLETIN BOARDS

SYDNEY:

INFOCENTRE	02 344 9511
TANDY ACCESS	02 625 8071
THE COCOCONNECTION	02 618 3591

HOBART

HUB	002 49 4405

VIDEOTEX SYSTEMS

VIATEL	01995
MOUSETEX	059 42 5528
VIX 4000	03 741 3295

TANDY INFO ON VIATEL

GUIDLINK	VIATEL #642#
PARIS RADIO	VIATEL #64268#
POWER CODE	VIATEL #64265#
TANDY	VIATEL #64261#

SOME TANDY USERS ON VIATEL

ALLAN BEALE	726353300
FRED BISSELING	648232630
JACK FRICKER	726268690
DEON GEORGE	753517750
JOHN GRIGSBY	945872030
STUART HALL	939765790
BOB KENNY	665122050
JEFF LARSEN	705471270
IAIN MACLEOD	944821350
CHRIS NAGLE	689523360
RICHARD PANKHURST	280717870
ROSS PRATT	648230650
RICCAY SCHMAHL	298151500
ARTHUR SLADE	262269400
DARCY O'TOOLE	755105770
RON WRIGHT	352924510