

THE Magazine for experienced InterTAN Color Computer Users !

\$4⁵⁰

AUSTRALIAN

COCO

MAGAZINE



This month's special features include:

The winners of the CoCo3

Program Competition and
the Music Competition.

Feature articles on Forth and OS9

plus

programs for Utilities, Graphics,

Games and Music.

VOL3N07
MARCH87

BETTER FOR MICROS



Now there's an exciting new world for Personal Computer owners to explore. The world of Goldlink 642 on Telecom Viatel.

All you need is a 1200/75 baud modem, the appropriate software, and a telephone line, and your PC will be ready to go.

Suddenly you'll be able to shop for software on your PC, and actually download* it directly through the Viatel system. You'll be able to get PC advice and tips. Even place messages on the system for other Viatel users to read and respond to — literally a PC talkback service that lets you have a say on almost any subject.

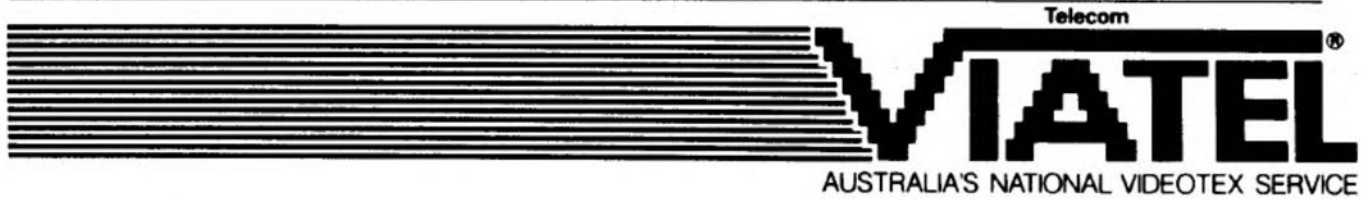
That's just part of what Goldlink 642 offers. And Goldlink 642 is just part of what Telecom Viatel offers. You can also bank with Viatel, place bets, buy and sell shares, book travel, and much more. Instantly, easily, economically. 24 hours a day.

Ask for a free brochure at any Telecom Business Office. And start using your micro in a whole new, better way.

* Coming



Telecom Australia
Better for Business



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 _____

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.

Utilities

7 SCREEN DUMPING
PMODE 4 screen dumping
for TANDY DMP printers.

8 DISASSEMBLER
Learning to understand,
write and adapt assembly
machine code programs.

10 DONKEY
Just what the Doctor
ordered

12 SECTOR BLASTER
Getting the most out of
your diskettes.

13 HIDE
Outta sight, but not outta
memory.

16 COCO3 UTILITIES
Fred alters and updates
a Public Domain utility.

17 SUPER INTELLIGENT
SCREEN DUMP
Not one but FOUR dumps
in one great program.

22 MEMORY LOOK
Taking a good, hard look at
memory on the graphics screen

23 COCO3 DRAWING
MACHINE
COCO3's graphics made
easy.

25 DISK FIX FOR COCO3
Setting up Double Sided
Drives and increasing
their speed.



inside COCO

28 JOYMEM
Skipping through RAM with
a joystick.

28 LOWERCASE
Lowercase with no hardware
modifications needed.

Music

53 MUSIC
More music additions.

Graphics

14 TURTLE GRAPHICS
Looking at CoCo3 from a
different angle.

30 THE CoCo3 PROGRAM
COMPETITION WINNER
Seeing is believing!!!

37 COLOUR PLUS
Putting CoCo3's palette
to work.

40 PAST and PRESENT
COCO
Another CoCo3 Program entry.

Games

42 TANK 11
A NEW version of last
year's games competition.

44 OBSTACLE
Try a little light relief
with Craig Stewart's latest
effort.

Applications

22 BASE GENERATOR
converting numbers from
one base to another.

62 LIFE
Unsophisticated but with
some interesting patterns.

Business

46 COLDATUM
Clive Winsall's entry in
the CoCo3 Program Comp.

52 DATABASE
A great example of
adapting and enhancing.

54 CASHBOOK
Ian Lobley's prize winning
OS9 program.

Hardware Mod.

60 COLOR and PMODE3/4
CHANGER
Switching from PMODE3
and 4 with the flick of
the wrist.

Plus:- In a Nut Shell, Forth, OS9, Mach. Lang. Programming, Frickers Follies.

Australian CoCo Magazine is a copyright
publication of GOLDSOFT, P.O. Box 1742,
Southport, Qld. 4215. All articles and
programs are the property of their authors
and may only be copied for the purpose of
providing two backups to the magazine
purchaser.

Founder: Greg Wilson.
Managing Editor: Graham Morphet.
Editor: Alex Hartman.
Accounts: Annette Morphet.
Production: Julie Vidler.
Advertising: Graham Morphet.
Art: Jim Bentick.
Subeditors: John Redmond, Fred
Bisselling, Jack Fricker.

Special thanks to: Geoff Fiala,
Martha Gritwhistle, Mike Turk,
Brian Dougan.
Deadlines: 7th of preceeding month.
Phone: 075 51 0577 or contact on
Viatel *64213#
Registered Publication QBG 4009.

BLAXLAND COMPUTER SERVICES PTY. LTD.

134 GREAT WESTERN HIGHWAY,
BLAXLAND. N.S.W. 2774

TELEPHONE: (047) 39 3903

HARD DRIVES

5 MG Hard Drive for
CoCo with controller \$1000.00

10 MG and 20 MG also
available

UPGRADES

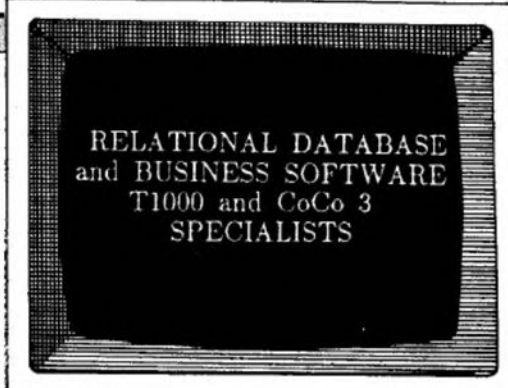
512 K Upgrades \$230.00
(Socketed from top,
won't work loose.)

PRINTER PAPER

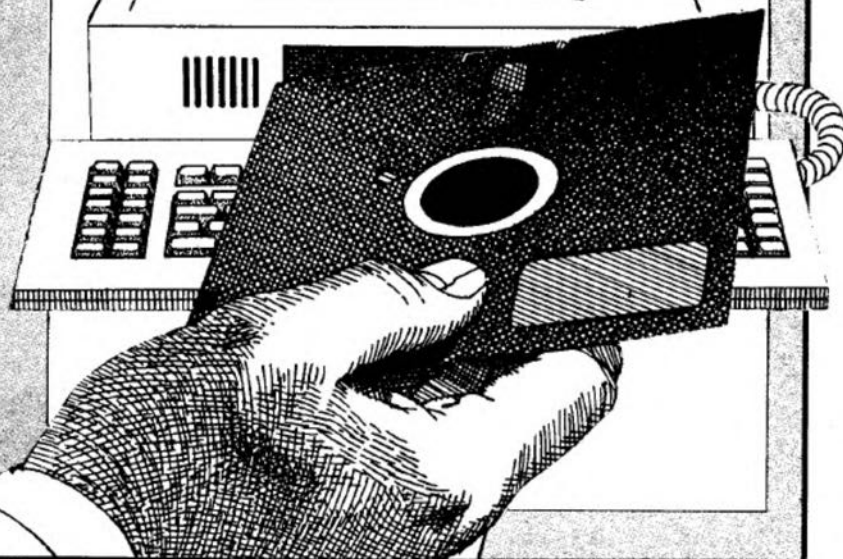
per 1000 \$17.00
per box of 4000 \$60.00

PRO DESIGN 11
(CAD Package for T1000)

SPECIAL PRICE THIS MONTH



RELATIONAL DATABASE
and BUSINESS SOFTWARE
T1000 and CoCo 3
SPECIALISTS



Tandy

ELECTRONICS DEALER SILICON CRAFTS

SOUTH WINDSOR
SHOP 1 / 499 GEORGE ST.,
(045) 77 6722

RICHMOND
14 BOSWORTH ST.,
(045) 78 4101

LIMITED STOCK OFFER ON COLOUR COMPUTER 3

FOR ENTERTAINMENT AND EDUCATION

SAVE

\$150



\$ 299⁹⁵

● Now with razor-sharp 640 x 200 graphics

● Connect to your own TV or to a Hi-Res monitor for brilliant graphics

DELIVERY EXTRA



GOLOLINK COM.

STATION

642 on VIATEL

SPECIAL OFFER: LIMITED STOCK

PRODUCTS		R.R.P.	OUR PRICE
TARGET PLANCALC	26-1512	\$ 49.95	\$ 29.95
PFS-FILE	26-1518	69.95	17.95
PROFILE III PLUS	26-1592		17.95
ALPHA KEY	26-1718		17.95
DELUX R232P.PAK	26-2226	179.95	99.95
COLOR LOGO ROM	26-2722	89.95	59.95
COL.ROBOT BATTLE	26-3070	69.95	44.95
REACTOIDS	26-3092	49.95	29.95
CC SPCH/SOUND	26-3144	189.95	99.95
COLOR PROFILE	26-3254	119.95	79.95
CC EDIT ASSEMBLR	26-3250	99.95	59.95
DELUX JOYSTICK	26-3012	49.95	34.95
MODEL 4 DISK-DRIVE	SPECIAL PRICE		CALL
CGP-115 PRINTER	26-1192		99.95

CALL SILICON CRAFTS AT (045) 78 4101
OR (045) 77 6722

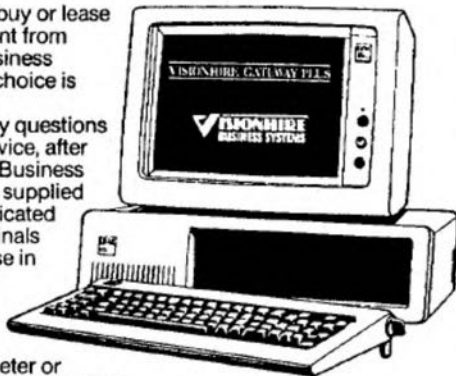
VIDEOTEX EQUIPMENT SOLUTIONS

Anyone with a telephone can access Videotex and Visionhire Business Systems will tailor the hardware and software to suit your needs, ranging from an adaptor for only \$18.00 per month up to IBM's PC of the year, the AT. Only the latest equipment from famous brands like IBM, Apple, Commodore, Tandata and Sony is used, all backed by Visioncare's Australia wide full service network.

You can rent, buy or lease your equipment from Visionhire Business Systems, the choice is yours.

If you have any questions just ask for advice, after all, Visionhire Business Systems have supplied 75% of all dedicated videotex terminals currently in use in Australia.

For friendly advice ring Will Ballschieter or Don Sanderson on 525723.



**VISIONHIRE
BUSINESS SYSTEMS**
Division of Visioncare Australia Pty Ltd

36 Brookes Street, Bowen Hills

Hardware/Software Specialists

For All Your CoCo Needs

AUTO ANSWER \$399.00

INFO CENTRE

THE FIRST BULLETIN BOARD SYSTEM

for Tandy's computers

(02) 344 9511 — 300 BPS (24 Hours)

(02) 344 9600 — 1200/75 BPS

(After Hours Only)

SPECIAL!

Avtek Mini Modem + Cable + CoCo
Tex Program — the total Viatel System —
\$279.00

We also have the largest range of Software for
OS-9 and Flex operating systems.

PARIS RADIO ELECTRONICS

161 Bunnerong Rd Kingsford N.S.W. 2032

(02) 344 9111



DON'T FORGET!

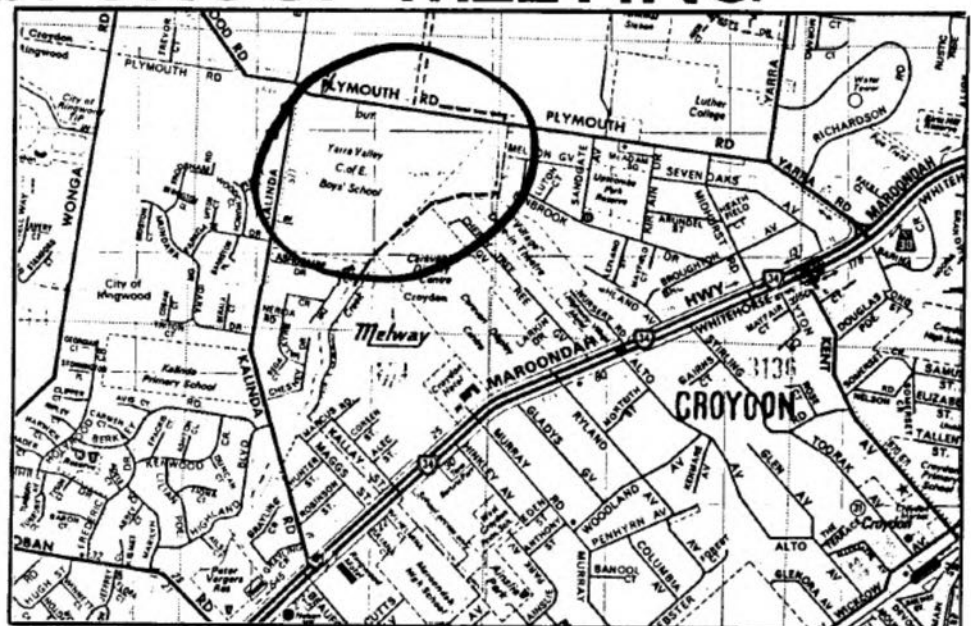
- OS9 Level 2
- Viatel
- Telgraf
- Question time

- Ron Wright
- Jerome Siapp
- Graham Morphett

USER GROUP MEETING

WEDNESDAY
the 11th
7PM

**YARRA
VALLEY
HIGH
SCHOOL**



See 'In a Nut Shell' for more details

IN A NUT SHELL



Welcome to Australian CoCo Magazine - the all Australian magazine for experienced InterTAN Colour Computer Users!

This month we present a wide sampling of programs by Australians which prove that we can produce material equivalent to the best the world has to offer!

We also announce the winners of the two competitions we've been running! And some pretty brilliant material has come in for these competitions too!

Next, we announce the Grafix competition! Last year we received heaps of grafix for this competition and we expect to be inundated this time too!

There will be two sections to the competition - a CoCo 2 section and a CoCo 3 section.

There will also be a special prize for an outstanding ANIMATED scene, and a beginners' prize for those who have owned their CoCo for less than 12 months (proof required!!).

So get your thinking caps on!! Let's see what you can do with your computer's grafix capabilities!! The competition closes on 31st July, 1987!

Speaking of Grafix, an ability available to CoCo 3 users is the ability to fade pictures & characters in and out. It's early days yet of course, but we haven't seen too many programs which use this feature - it might be a nice feature to contemplate for your entry in the Grafix Competition!

And to answer the question of the older readers, yes, the judge is a little biased this year too!

She would prefer landscapes or space scenes - but especially landscapes!

However, if you have a great idea for a Grafix, don't worry about her biases - if it's good, we'll make sure the judge considers your entry seriously!

The Judge? Oh sorry! It's Martha G. herself. (Sorry about that, but she's the only one here with the time available!)

On the subject of competitions, don't forget the BIG ONE! The Games Competition this year is really big news!

Get your entry in by 31st July, 1987 and you could have your program purchased by InterTAN for sale in their stores this Christmas!

But don't forget too that programs in other categories also win prizes - and all prizes will be presented at Conf '87 to be held in Sydney in August on a date yet to be fixed!

Graham will be in Melbourne on the week of 11th March.

He will be at the Ringwood Users' Group meeting on the night of 11th March. This is held at the Yarra Valley High School and will start at 7.30pm sharp. There will be a \$1 entry fee to cover the cost of room hire and supper.

A map is included later in this magazine to show you how to get there.

Topics to be covered at the meeting will include Viatel, Telgraf, 512K upgrades to the CoCo 3, OS 9 and a question and answer session.

Last time we did this, we had a lot of people, so come early to get a seat!

Later in the same week, there will be two 'bashes' (Viatal parties), details of which will be available through the office here at Goldsoft, or through Andrew Rawlings, or from Goldlink on Viatel.

Come along! You'll have a ball!

Telgraf Videotex is a new Videotex System which will be online this month.

As Telgraf is new it will therefore take a little while to build, however within a few months it will have the following facilities available:

- * National & International Telexing - send & receive;
- * 12000 page State Service Directory - similar to Yellow Pages;
- * 100000 page encyclopedia;
- * Complete Dictionary;
- * Financial Reports;
- * On line Tutor;
- * Complete Education (Yrs 1 to 12);
- * Public National Data Mart (168000 items);
- * NO ON LINE CHARGES!!
- * Access from every STD area;
- * State Databases for information on your state;
- * Telesoftware;
- * Home Banking;
- * A visitor's database.

The system will feature a 0c / minute line time charge - a very big advantage. And Goldlink will be online from day one on node *6420#.

We don't see this system as a replacement for Viatel, but rather an adjunct, and we'll be growing Goldlink on both systems according to individual demands.

So now is the time to get cracking and purchase that modem and videotex software for your computer!

If you need information, the information is on videotex!

Faster than a speeding ticket....

SCREEN DUMPING

16K ECB + DMP Printer

by Craig Stewart

SCREEN DUMP ... THIS IS yet another PMODE 4 screen dump that should work on all Tandy DMP printers, but with a bit of a difference. I have seen many screen-dumps, both BASIC (very slowww) and M/L, but all have, it seems to me, one problem - they only dump the screen on one corner of the page, this being not much good for title pages, etc. (although I think Tandy sell one similar to mine).

Answer - simply double the printout size -- new problem - DMP printers have only 480 graphics dots across the page ... 256 doesn't go into 480 too well.

Therefore, SCR-DUMP has to cut off the first 8 pixels from each side of the screen.

Another problem I had to face, was that DMP printers only take up 7 bits at a time. Since each pixel is now 2 bits, this means there is half a pixel left over on every second pass. Although this was reasonably easily solved, it has lengthened the M/L to quite a bit more than (say) a normal M/L dump.

Nonetheless, it works fine, and being written in M/L is very fast - being only limited by the printer baud-rate. (eg with my DMP-130 printer working at 2400 baud (POKE150,18), it takes less than 5 minutes to dump the entire screen.)



M. Hawkins T. Harris T. Earles

Sample of Screen-Dump, done on a DMP-130.
Entire picture took about 5 minutes to dump.

```
0 GOTO10
1 '***** SCREEN DUMP *****
  ***** CRAIG STEWART *****
3 SAVE"146A:3":END
10 CLS:PRINT "loading data .. pl
  ease wait..
20 LI=50:FOR X= 15824 TO 16121
STEP 70:LI=LI+10:T=0
30 READ T$:FOR Y=X TO X+69:READ
A$:A=VAL("&H"+A$):T=T+A:POKE Y,A
:NEXT Y:IF HEX$(T)=T$ THEN 40
35 PRINT "error in line ";LI:END
40 NEXT X:READ T$:T=0:LI=LI+10:F
OR Z=X-70 TO 16121:READ A$:A=VA
L("&H"+A$):T=T+A:POKE Z,A:NEXT Z
:IF HEX$(T)<>T$ THEN 35 ELSE PRI
NT:PRINT"data loading complete"
42 PRINT"IS PRINT-DUMP FOR A (D)
ISK OR A (C)ASSETTE SYSTEM"
43 A$=INKEY$:IF A$="D" THEN 45 E
LSE IF A$="C" THEN POKE16109,6 E
LSE GOTO 43
45 PRINT "press enter to save":L
INEINPUTZ$
50 SAVEM "DUMP16K", 15824, 16121
, 15824 :END
60 DATA 1BAF,86,FE,B7,0,6F,86,D,
AD,9F,A0,2,86,12,AD,9F,A0,2,7F,3
E,FA,86,8,B7,3E,F9,17,0,C5,86,80
,17,0,CA,27,2,8B,3,7C,3E,FA,17,0
,CO,27,2,8B,C,7C,3E,FA,17,0,B6,2
7,2,8B,30,7C,3E,FA,17,0,AC,27,B,
8B,40,F6,3E,F9
```

```
70 DATA 22E1,8E,3E,FE,3A,E7,84,A
D,9F,A0,2,AD,9F,A0,2,B6,3E,FA,80
,3,B7,3E,FA,B6,3E,F9,4C,B7,3E,F9
,81,F8,25,B5,86,D,AD,9F,A0,2,86,
8,B7,3E,F9,B6,3E,FA,8B,4,B7,3E,F
A,86,80,8E,3E,FE,F6,3E,F9,3A,6D,
84,27,2,8B,1,17,0,5F
80 DATA 1D7A,27,2,8B,6,7C,3E,FA,
17,0,55,27,2,8B,18,7C,3E,FA,17,0
,4B,27,2,8B,60,AD,9F,A0,2,AD,9F,
A0,2,B6,3E,FA,81,BE,22,21,80,2,B
7,3E,FA,B6,3E,F9,4C,B7,3E,F9,81,
F8,25,B7,B6,3E,FA,8B,6,81,BF,22,
8,80,3,B7,3E,FA,16
90 DATA 2195,FF,40,86,D,AD,9F,A0
,2,86,1E,AD,9F,A0,2,39,5F,8E,3E,
FE,6F,80,5A,26,FB,39,34,2,8D,A,3
5,2,E6,84,FA,3E,FB,E1,84,39,B6,3
E,F9,C6,20,3D,B7,3E,FC,86,8,3D,8
E,3E,F0,30,86,E6,84,F7,3E,FB,B6,
3E,FA,C6,20,3D,1F,1,F6
100 DATA 18A2,3E,FC,3A,30,89,E,0
,39,80,40,20,10,8,4,2,1,0,0,D9,3
5,6,34,6,9E,C3,9F,BD,9E,C5,9F,BF
,CE,9E,7B,84,1,27,3,50,CB,8,58,5
8,33,C5,34,40,BD,9F,A7,35,40,33,
5E,34,10,BD,9F,A7,35,20,A6,E4,84
,3,27,6,81,3,27
110 DATA 373,3E,FC,3A,30,89,E,0
,39,80,40,20,10,8,4,2,1,0,0
```

Start learning to understand,
write and adapt assembly and
machine code programs.

DISASSEMBLER

32K CB
UTILITY

by Wilfred Tritscher

TURING PROVED, in the abstract concept, that all computers (Turing machines) are the same. Therefore it is not unreasonable to assume the converse corollary also applies. If you've learnt one programming language you've learnt them all.

But in real world limitations (usually in the area of "out of reach", financially speaking, software) stifle the practical realization of Turing's result.

I'm referring to learning assembler after mastering Color Basic. Barden's Color Computer Assembly Language Programming seems a good buy for \$12.95 but it is really only documentation for EDTASM+.

To get round the lack of an assembler I used the "DISASSEMBLER" (after conversion to CB) from Mar '86 Australian RAINBOW to decipher the Detailed Instruction Set in Appendix II of CCALP as well as any hex listings I had POKED into memory (LOADING and RUNNING "DISASSEMBLER" doesn't destroy high memory locations in a 16K machine).

"UTILPACK" (also in the same issue) was used to edit hex data strings (less typing, no commas) in BASIC programs, since CB has no EDIT command.

I realize this is a primitive way of getting into assembly language but it was enough to get a handle on understanding, writing (simple) and more importantly, adapting assembly and machine code programs.

Also since most hardcopy assembly listings include the assembled hex code with the program listing it probably involves less typing than the actual assembly mnemonics (OK, debugging is more difficult).

```
0 '6809 DISASSEMBLER FOR THE COC
0 BY ROD HOSKINSON
  REVISION 1.1:31/1/1986
  PUBLIC DOMAIN PROGRAM.
1 'MODIFIED TO RUN UNDER COLOR B
  ASIC BY W. TRITSCHER. LINES 4 TO
  7 INCLUSIVE INCORPORATE A SCREE
  N COLOUR REVERSAL UTILITY WHICH,
  SINCE I USE A B&W TV, HAVE NOT
  BEEN CONVERTED TO CB.
2 CLS:POKE65495,0:GOTO8
3 POKE65494,0:SAVE"147A:3":END
4 CLEAR300,9833:RESTORE:M=9834:F
  ORI=1TO3:READS,D:FORJ=S TOD:POKE
  M,PEEK(J):M=M+1:NEXTJ,I
5 DATA33395,33413,38316,38350,41
  738,41825
6 FORI=1TO25:READM$,D$:M=VAL("&H
  26"+M$):D=VAL("&H"+D$):POKEM,D:N
  EXT
7 DATA6E,0,6F,D,9A,F,9F,16,F4,16
  ,F5,32,F6,62,F7,39,AE,20,BA,20,A
  F,8D,B0,48,CF,8,D2,22,D3,2,D4,88
  ,D5,40,D6,84,D7,DF,EF,20,F9,A7,F
  A,84,FB,A7,FC,82,FD,39
8 S=0:E=0:DN=0:PRINT"MC6809E di
  ssembler FOR COCO BY ROD H
  OSKINSON V1.1 1986 (COLOR BASI
  C V. BY W. TRITSCHER)":PRINT
10 PRINT"DISASSEMBLE ASCII DUMP
  BYTE DUMPWORD DUMP aLOADM eND
  ":IFDN=-2THENPRINT"SCREEN ";
  ELSEPRINT"PRINTER ";
11 IFS<OORS>65535ORE<OORE>65535T
  HENS=0:E=0ELSEIFS>E THENMS=E
12 PRINT"PRESS I THEN d OR a,b,w
  ,c,e,s,p":ZN=S:GOSUB256:HX=E:GOS
  UB276:PRINT"START=":RIGHT$(  
"0000
  "+ZH$,4):" END="RIGHT$(  
"0000"+HX
  $,4):" *";
13 AS=INKEY$:IFAS=""THENEXEC4136
9:GOTO13
14 IFAS="D"THEN16ELSEIFAS="A"THE
  N16ELSEIFAS="W"THEN167ELSEIFAS=
  "C"THEN170ELSEIFAS="B"THEN172ELS
  EIFAS="P"THEN179ELSEIFAS="S"THEN
  180ELSEIFAS="I"THENGOSUB17:GOTO1
  2
15 IFAS="E"THENCLS:POKE65494,0:E
  ND:ELSE13
16 PRINT"DISASSEMBLE":GOTO25
17 PRINT:PRINT"START=":INPUTC$
18 PRINT"END=":INPUTD$
19 IFRIGHT$(C$,1)="H"THENZH$=LEF
  T$(C$,LEN(C$)-1):GOSUB280:S=ZN E
  LSES=VAL(C$):THE OLD Z-80 EDTAS
  M HEX FORMAT ddddH IS USED
20 PO=S
21 IFRIGHT$(D$,1)="H"THENZH$=LEF
  T$(D$,LEN(D$)-1):GOSUB280:E=ZN E
```

```
LSEE=VAL(D$):'FOR COCO EDTASM+ F
ORMAT CHANGE "H" TO "$" AND RIGH
T$ TO LEFT$ AND LEFT$ TO RIGHT$
IN LINES 19 AND 21
22 IFS<S THEN17
23 IFS<OORS>65535ORE<OORE>65535T
HEN17
24 RETURN
25 Z=0:PO=S:BG=S:ST=S:G$="" :B=PE
EK(S):F=0
26 IFB<>16 AND B<>17 THENMS=S+1:G
OTO34
27 T=PEEK(S+1):IFT>32 ANDT<48 TH
ENS=S+2:B=T:F=1:GOTO34
28 IFB=17 THENGOSUB206:C=50 ELSE
GOSUB205:C=155
29 FORA=1TOC STEP7:Y$=MID$(S$,A,
7):ZH$=LEFT$(Y$,2):GOSUB280
30 IFZN<>T THEN32ELSEMS=MID$(Y$,
3,1):G$=RIGHT$(Y$,4):IFRIGHT$(G$
,1)=" "THENMS=LEFT$(G$,3)
31 S=S+2:GOTO43
32 NEXT
33 S=S+2:G$="???":GOTO149
34 Q=INT(B/16)+1:ONQ GOSUB193,19
4,195,196,197,198,199,200,201,20
2,202,202,204,203,203,203:G$=MID
$(S$, (B-(Q-1)*16)*5+1,5)
35 IFB<144 OR (B)<191 AND B<208)
THENMS=RIGHT$(G$,1):G$=LEFT$(G$,
4):IFM$=" "THENMS=RIGHT$(G$,1):G
$=LEFT$(G$,3)
36 IFB<144 OR (B)<191 ANDB<208) T
HEN40
37 G$=LEFT$(G$,4):IFRIGHT$(G$,1)
=" "THENMS=LEFT$(G$,3)
38 M=0:M=-68*(B)<143 ANDB<160)-68
*(B)<207 ANDB<224)-73*(B)<159 ANDB
<176)-73*(B)<223 ANDB<240):IFM=0T
HENMS="E"ELSEMS=CHR$(M)
39 GOTO41
40 IFF=1THENG$="L"+G$
41 IFG$="***"THENG$="???":GOTO1
49
42 IFG$="----"THENG$="ANDCC":M$=
"1"
43 IFM$="H"THEN149
44 IFM$<"R"THEN55
45 G$=G$+" $"
46 IFLEFT$(G$,1)="L"THENB=16ELSE
B=8
47 IFB=16THENT=PEEK(S)*256+PEEK(
S+1):S=S+2:ELSET=PEEK(S):S=S+1
48 IFB=8THEN52
49 IFT>32767THENT=- (65535-T)
50 GOSUB182:ZN=S+T:GOSUB256:T$=Z
H$:GOSUB157
51 GOTO54
52 IFT>127THENT=- (256-T)
53 GOSUB182:ZN=S+T:GOSUB256:T$=Z
H$:GOSUB157
54 G$=G$+T$:GOTO149
```

```

55 IFM$<"D" THEN61
56 G$=G$+" <$"
57 ZN=PEEK(S):GOSUB256:T$=ZH$:GO
SUB158
58 S=S+1
59 G$=G$+T$
60 GOTO149
61 IFM$<"E" THEN64
62 G$=G$+" S"
63 ZN=PEEK(S)*256+PEEK(S+1):GOSU
B256:T$=ZH$:GOSUB17:S=S+2:G$=G$
+T$:GOTO149
64 IFM$="1"ORM$="2" THEN65ELSE71
65 G$=G$+" #S"
66 IFM$="1" THENZN=PEEK(S):GOSUB2
56:T$=ZH$:S=S+1:GOSUB158
67 IFM$="2" THENZN=PEEK(S)*256+PE
EK(S+1):GOSUB256:T$=ZH$:S=S+2:GO
SUB157
68 G$=G$+T$
69 IFLEFT$(G$,4)="ORCC" OR LEFT$
(G$,5)="ANDCC" THENGOSUB184:GOTO1
49
70 T$=ZH$:GOSUB280:IFM$="1" AND
ZN<127 AND ZN>31 THENG$=G$+" "+CH
R$(ZN):GOTO149ELSE149
71 IFM$<"M" THEN102
72 G$=G$+" "
73 IFLEFT$(G$,3)="PSH"ORLEFT$(G$
,3)="PUL" THEN90
74 B=PEEK(S):S=S+1
75 ZN=B:GOSUB256:T$=ZH$:GOSUB158
76 L$=LEFT$(T$,1):R$=RIGHT$(T$,1
):IFASC(L$)>64 THENL=ASC(L$)-55EL
SEL=VAL(L$)
77 IFASC(R$)>64 THENR=ASC(R$)-55E
LSER=VAL(R$)
78 N=L:GOSUB79:G$=G$+T$+"":N=R:
GOSUB79:G$=G$+T$:GOTO149
79 IFN=0 THENT$="D":RETURN
80 IFN=1 THENT$="X":RETURN
81 IFN=2 THENT$="Y":RETURN
82 IFN=3 THENT$="U":RETURN
83 IFN=4 THENT$="S":RETURN
84 IFN=5 THENT$="PC":RETURN
85 IFN=8 THENT$="A":RETURN
86 IFN=9 THENT$="B":RETURN
87 IFN=10 THENT$="CC":RETURN
88 IFN=11 THENT$="DP":RETURN
89 T$="???":RETURN
90 D=PEEK(S):S=S+1
91 IF(D AND128)=128 THENG$=G$+"PC
,"
92 IF(D AND64)=64 ANDLEFT$(G$,1)=
"U" THENG$=G$+"S":GOTO94
93 IF(D AND64)=64 THENG$=G$+"U,"
94 IF(D AND32)=32 THENG$=G$+"Y,"
95 IF(D AND16)=16 THENG$=G$+"X,"
96 IF(D AND8)=8 THENG$=G$+"DP,"
97 IF(D AND4)=4 THENG$=G$+"B,"
98 IF(D AND2)=2 THENG$=G$+"A,"
99 IF(D AND1)=1 THENG$=G$+"CC,"
100 IFRIGHT$(G$,1)="," THENG$=LEF
T$(G$,LEN(G$)-1)
101 GOTO149
102 D=PEEK(S):S=S+1
103 PM$=""
104 OS="" : ES="" : F=0
105 IF(D AND16)=16 AND (D AND128
)=128 THENF=16:D=D-16
106 IFD=159 THENZN=PEEK(S)*256+PE
EK(S+1):GOSUB256:G$=G$+" "+["$"+
RIGHT$("0000"+ZH$,4)+"1":S=S+2:G
OTO149
107 Q=D AND31
108 IF(D AND128)<0 THEN112
109 T=-16*(D AND16)=16)-8*(D A

```

```

ND8)=8)-4*(D AND4)=4)-2*(D AND
2)=2)-1*(D AND1)=1)
110 IFT>15 THENT=- (32-T)
111 OS=STR$(T):GOTO130
112 IFQ=4 THEN130
113 IFQ<8 THEN116
114 T=PEEK(S):S=S+1:IFT>127 THENT
=- (256-T)
115 OS=STR$(T):GOTO130
116 IFQ<9 THEN119
117 T=PEEK(S)*256+PEEK(S+1):S=S+
2:IFT>32767 THENT=- (65536-T)
118 OS=STR$(T):GOTO130
119 IFQ=6 THENOS="A":GOTO130
120 IFQ=5 THENOS="B":GOTO130
121 IFQ=11 THENOS="D":GOTO130
122 IFQ=0 THENPM$="" :GOTO130
123 IFQ=1 THENPM$="" ++:GOTO130
124 IFQ=2 THENPM$="" -:GOTO130
125 IFQ=3 THENPM$="" --:GOTO130
126 IFQ=12 THENT=PEEK(S):ES="PCR"
:S=S+1:GOSUB182:IFT>127 THENT=- (2
56-T)
127 IFQ=12 THENZN=S+T:GOSUB256:OS
=" $" +RIGHT$("0000"+ZH$,4):GOTO13
0
128 IFQ=13 THENES="PCR":T=PEEK(S)
*256+PEEK(S+1):S=S+2:GOSUB182:IF
T>32767 THENT=- (65536-T)
129 IFQ=13 THENQ=12:GOTO127
130 IFLEFT$(OS,1)=" " THENOS=MID$
(OS,2)
131 GS=G$+" "+OS
132 IFE$="PCR" THENG$=G$+" "+ES:G
OTO143
133 T=D AND96
134 IFT=0 THENES="X"
135 IFT=32 THENES="Y"
136 IFT=64 THENES="U"
137 IFT=96 THENES="S"
138 IFPM$="" + THENES=ES+"+"
139 IFPM$="" ++ THENES=ES+"++"
140 IFPM$="" - THENES="" - +ES
141 IFPM$="" -- THENES="" -- +ES
142 GS=G$+" "+ES
143 IFF=0 THEN149
144 FORA=1 TOLEN(G$)
145 IFMID$(G$,A,1)=" " THEN147
146 NEXT
147 T$=LEFT$(G$,A):Y$=MID$(G$,A+
1)
148 G$=T$+"[" +Y$+"]"
149 ES="" : FORA=ST TO S-1
150 ZN=PEEK(A):GOSUB256:T$=ZH$:G
OSUB158
151 ES=ES+T$:NEXT
152 ZS=G$:ZT$="" :GOSUB264:J=ZI
:IFJ=0 THEN153 ELSESEQ$=MID$(G$,J+1)
:ZL=6-J:ZC=32:GOSUB272:G$=LEFT$(
G$,J)+ZS+Q$
153 POKE65494,0:ZL=11-LEN(ES):ZC
=32:GOSUB272:ZN=PO:GOSUB256:PRIN
T#DN,RIGHT$("0000"+ZH$,4):" ";ES
;ZS;G$:POKE65495,0
154 IFINKEY$="" THEN12
155 PO=PO+S-PO
156 IFS=E THEN25 ELSE12
157 T$=RIGHT$("0000"+T$,4):RETUR
N
158 T$=RIGHT$("00"+T$,2):RETURN
159 RETURN
160 PRINT"ASCII DUMP":DISASSEMB
LES FCC TABLES
161 FORS=S TOE
162 ZN=S:GOSUB256:HX=PEEK(S):GOS
UB276:T$=RIGHT$("0000"+ZH$,4)+"
"+RIGHT$("00"+HX$,2)+" " F

```

```

CC "
163 Q=PEEK(S):IFQ>31 AND Q<128TH
ENT$=T$+CHR$(Q):GOTO166
164 IFQ>159 THENENT$=T$+"$80+" +CH
R$(Q-128):GOTO166
165 T$=T$+"???"
166 POKE65494,0:PRINT#DN,T$:POKE
65495,0:IFINKEY$="" THEN12ELSENE
XT:GOTO12
167 PRINT"WORD DUMP":DISASSEMBL
ES FDB TABLES
168 FORS=S TOE STEP2:ZN=S:GOSUB2
56:HX=PEEK(S)*256+PEEK(S+1):GOSU
B276:T$=RIGHT$("0000"+ZH$,4)+" "
+RIGHT$("0000"+HX$,4)+" " FD
B $" +RIGHT$("0000"+HX$,4)
169 POKE65494,0:PRINT#DN,T$:POKE
65495,0:IFINKEY$="" THEN12ELSENE
XT:GOTO12
170 PRINT"CLOADM PROGRAM":INPUT"
PROGRAM NAME":N$:INPUT"LOADING O
FFSET":LO:POKE65494,0:CLOADM N$,
LO:POKE65495,0:THE CURSOR VANIS
HES AND NO "S" APPEARS WHEN CLOA
DM IS USED WITHIN A PROGRAM
171 ZN=PEEK(487)*256+PEEK(488):G
OSUB256:HX=PEEK(126)*256+PEEK(12
7)-1:GOSUB276:PRINT"START ADDRESS
S=$":RIGHT$("0000"+ZH$,4):PRINT"
END ADDRESS=$":RIGHT$("0000"+HX$
,4):ZN=PEEK(157)*256+PEEK(158):G
OSUB256:PRINT"EXECUTE ADDRESS=$"
:RIGHT$("0000"+ZH$,4):GOT
172 PRINT"BYTE DUMP":DISASSEMBL
ES FCB TABLES
173 FORS=S TOE
174 ZN=S:GOSUB256:HX=PEEK(S):GOS
UB276:T$=RIGHT$("0000"+ZH$,4)+"
"+RIGHT$("00"+HX$,2)+" " F
CB $" +RIGHT$("00"+HX$,2)
175 POKE65494,0:PRINT#DN,T$:POKE
65495,0
176 IFINKEY$="" THEN12
177 NEXT
178 GOTO12
179 DN=-2:PRINT:GOTO10:'HAVE NO
PRINTER CANNOT VOUCH FOR PRINTER
ROUTINE
180 DN=0:CLS
181 GOTO12
182 IF(S+T)<0 THENT=65536+T:RETUR
N
183 IF(S+T)>65536 THENT=T-65536:R
ETURNELSERETURN
184 ZH$=RIGHT$(G$,2):GOSUB280:Q=
ZN:W$=""
185 IF(Q AND128)=128 THENW$="E"
186 IF(Q AND64)=64 THENW$=W$+"F"
187 IF(Q AND32)=32 THENW$=W$+"H"
188 IF(Q AND16)=16 THENW$=W$+"I"
189 IF(Q AND8)=8 THENW$=W$+"N"
190 IF(Q AND4)=4 THENW$=W$+"Z"
191 IF(Q AND2)=2 THENW$=W$+"V"
192 IF(Q AND1)=1 THENW$=W$+"C":G$
=G$+"="+W$:RETURNELSEG$=G$+"="+V
$:RETURN
193 S$="NEGD *****COMD LSRD
*****RORD ASRD ASLD ROLD DECD *
***INCD TSTD JMPD CLRD ":RETURN
194 S$="*****NOPH SYNCH****
*****LBRARLSRR*****DAAH ORCCM*
***-----SEXH EXGM TFRM ":RETURN
195 S$="BRAR BRNR BHIR BLSR BHSR
BLOR BNER BEQR BVCR BVSR BPLR B

```

Make an ass out of yourself with...



DONKEY

32K ECB
UTILITIES

by Charles Bartlett

THERE ARE A LOT of good CoCo 2 programs that when converted to appear on the CoCo 3 high resolution screen are improved beyond belief, more if they were originally PMODE 4 programs in only two colors or artifact American programs. The problem is that going through them line by line adding a 'H' to the relevant commands can take hours. That's why I wrote DONKEY to do all the DONKEY work.

Loading

DONKEY only has to be loaded once, you can convert as many programs as you wish without reloading DONKEY. Firstly type:

```
CLEAR200,&H7F9F
```

Then for cassette systems type:

```
CLOADM"DONKEY"
```

or for disk systems type:

```
LOADM"DONKEY/BIN"
```

Now load in a copy of the Basic program you want to convert. Now type:-

```
EXEC
```

Depending on the length of your program there will be a pause of a few seconds.

The average length program takes about 3 to 4 seconds to convert.

After this pause the computer prompt will reappear. You can now save the program and load in another one for conversion if required.

What DONKEY does

DONKEY changes the following PMODE screen commands to the relevant HSCREEN commands.

The PMODE commands are:

```
CIRCLE, LINE, PAINT, PSET,  
SCREEN, PPOINT, PRESET, DRAW,  
COLOR, PCLS, GET, PUT
```

and the HSCREEN commands are:

```
HCIRCLE, HLINE, HPAINT, HSET,  
HSCREEN, HPOINT, HRESET, HDRAW,  
HCOLOR, HCLS, HGET, HPUT
```

DONKEY only changes the Basic tokens that it has been told to look for, so you will still have to make some MINOR changes. For example, before:

```
GET(10,10)-(20,20),A,G
```

... and after:

```
HGET(10,10)-(20,20),A,G
```

... and another example, before:

```
SCREEN1,1
```

... and after:

```
HSCREEN1,1
```

In the first example you still need to add the HBUFF command to the program and then change the end of the HGET line and in the second example you need to change the end of the HSCREEN command to select the required screen and delete the unwanted parameters.

DONKEY could not do this for you as it did not and could not know what buffer or screen you would require.

ALL the other commands that DONKEY has changed for you will require no further editing by you.

Some LOGIC changes may be required, as an example, one of the programs I used to test DONKEY was the LUDO program published in a recent edition. Only one line had to be changed by me after DONKEY had done its work before the program could be run. After DONKEY had done its job the line looked like this:-

```
52 PMODE4,1:HCLS:HSCREEN1,1
```

This was originally:

```
52 PMODE4,1:PCLS:SCREEN1,1
```

I then edited the line to read:

```
52 HSCREEN2:HCLS2
```

NO OTHER CHANGES WERE REQUIRED!

You might want to change the colors, but this is not connected with actually getting the program to run.

DONKEY lives at &H7FA0 to &H7FFF so if you run a converted program and it interferes with this memory area, you will have to reload DONKEY if you want to do more conversions.

DONKEY was hand compiled as I don't have an assembler, but a typed simulation of an assembly listing follows ...

More for Less

by Charles Bartlett

WITH THE FOLLOWING one liner you can get an extra 2K of memory to use from BASIC. This routine is adapted from the PCLEAR0 routine published a few issues ago. You might think of this adaptation as a PCLEAR-1.

The Basic pointers are moved right back to the beginning of the low resolution text screen. That is the reason for the WIDTH40 command. You can change to and from the 40 and 80 column screens after running this program, but you MUST NOT return to the 32 column screen or you will crash your computer and lose any resident Basic program.

After running this program your memory size response should be 31528 which is exactly 2K greater than the normal response that you would normally get after a PCLEAR1 which is 29480

The listing:

```
10 WIDTH40:FOR X=&H03B6 TO  
&H03BD:READ A$:POKE X,VAL(  
"&H"+A$):NEXT DATA CC, 04,  
01, 1F, 02, 7E, 96, A5:EXEC  
&H03B6:NEW
```

7FA0 BE0019		ORG #57FA0	
7FA3 201B		LDX \$0019	GET START OF BASIC PROGRAM
7FA5 C60C	LOOP	BRA SKIP	TO GET TO START OF FIRST LINE
7FA7 108E7FE8		LDB #50C	LOAD LENGTH OF TABLE
7FAB A680		LDY TABLE	POINT TO START OF TABLE
7FAD 8100		LDA ,X+	GET TOKEN FROM BASIC & INCREMENT POINTER
7FAF 270F		CMPA #500	IS IT END OF LINE
7FB1 A1A4	MORE	BEQ SKIP	GO IF YES
7FB3 270F		CMPA ,Y	LOOK IN TABLE FOR TOKEN
7FB5 3122		BEQ SWAP	IF FOUND THEN PROCESS
7FB7 5A		LEAY +2,Y	MOVE POINTER TO NEXT ITEM IN TABLE
7FB8 26F7		DECB	ADJUST COUNTER
7FBA BC001B		BNE MORE	DO AGAIN IF NOT END OF TABLE
7FBD 2FE6		CMPX \$001B	IS IT THE END OF BASIC PROGRAM
7FBF 39		BLE LOOP	CONTINUE IF NOT
7FC0 3004	SKIP	RTS	BACK TO BASIC
7FC2 20E1		LEAX +4,X	DONT MESS WITH THESE BYTES
7FC4 81BD	SWAP	BRA LOOP	GO BACK FOR MORE
7FC6 2710		CMPA #5BD	IS TOKEN PSET
7FC8 81BE		BEQ COMMA	GO IF YES
7FCA 270C		CMPA #5BE	IS TOKEN PRESET
7FCC 81A0		BEQ COMMA	GO IF YES
7FCE 2710		CMPA #5A0	IS TOKEN PPOINT
7FD0 A6A0	OK	BEQ POINT	GO IF YES
7FD2 A6A0		LDA ,Y+	JUST TO MOVE POINTER
7FD4 A71F		LDA ,Y+	GET NEW TOKEN
7FD6 20CD		STA -1,X	PUT NEW TOKEN IN BASIC PROGRAM
7FD8 A61E	COMMA	BRA LOOP	DO IT ALL AGAIN
7FDA 812C		LDA -2,X	GET BYTE BEFORE TOKEN
7FDC 27C7		CMPA #52C	IF ITS A COMMA, ITS ON THE END OF A LINE
7FDE 20F0		BEQ LOOP	COMMAND & WE DONT WANT TO CHANGE IT
7FE0 A61E	POINT	BRA OK	OK TO CHANGE IT
7FE2 81FF		LDA -2,X	GET BYTE BEFORE TOKEN. IF ITS NOT \$FF
7FE4 26BF		CMPA #5FF	THEN THIS IS A SOUND COMMAND.
7FE6 20E8		BNE LOOP	DONT CHANGE IT
7FE8 BC	TABLE	BRA OK	ITS PPOINT, SO CHANGE IT
7FE9 E6		FCB \$BC	TOKEN FOR PCLS
7FEA BB		FCB \$E6	TOKEN FOR HCLS
7FEB EA		FCB \$BB	TOKEN FOR LINE
7FEC BD		FCB \$EA	TOKEN FOR HLINE
7FED F3		FCB \$BD	TOKEN FOR PSET
7FEE BE		FCB \$F3	TOKEN FOR HSET
7FEF F4		FCB \$BE	TOKEN FOR PRESET
7FF0 BF		FCB \$F4	TOKEN FOR HRESET
7FF1 E4		FCB \$BF	TOKEN FOR SCREEN
7FF2 C1		FCB \$E4	TOKEN FOR HSCREEN
7FF3 E7		FCB \$C1	TOKEN FOR COLOR
7FF4 C2		FCB \$E7	TOKEN FOR HCOLOR
7FF5 E9		FCB \$C2	TOKEN FOR CIRCLE
7FF6 C3		FCB \$E9	TOKEN FOR HCIRCLE
7FF7 E8		FCB \$C3	TOKEN FOR PAINT
7FF8 C4		FCB \$E8	TOKEN FOR HPAINT
7FF9 EB		FCB \$C4	TOKEN FOR GET
7FFA C5		FCB \$EB	TOKEN FOR HGET
7FFB EC		FCB \$C5	TOKEN FOR PUT
7FFC C6		FCB \$EC	TOKEN FOR HPUT
7FFD F5		FCB \$C6	TOKEN FOR DRAW
7FFE A0		FCB \$F5	TOKEN FOR HDRAW
7FFF AB		FCB \$A0	TOKEN FOR PPOINT
		FCB \$AB	TOKEN FOR HPOINT

Getting rid of those disk nasties.

SECTOR BLASTER

64K DECB
UTILITY

by Alex

THE IDEA OF "Sector Blaster" came along some time ago when a friend of mine tried to format one of his slightly-used disks and found, after he had tried to DSKINI it, an I/O error.

Well what could you do? For one you could try to DSKINI it again. So he did and again the same thing happened - I/O error.

Normally you are restricted in what you could do with a disk that was already slightly used.

Either you had to 'save around' the corrupted sector or throw the whole disk away (disks are expensive and not everyone is going to throw away a disk that is only slightly cactus).

Well, no more of that. "Sector Blaster" will fix that problem, permanently (unless you DSKINI it!).

"Sector Blaster" uses "Disk Verifier" (Australian Rainbow Magazine, pg 41) to find the faulty sectors of your disk and uses additional BASIC code to 'blot out' the trouble-making sectors.

The result? Whenever you tell Disk BASIC to save a program, it will ignore that particular sector and save the rest of the program on the disk.

Again, this software does do funny things to BASIC. I suggest that when you finish using this program, you could start your CoCo.

Sometimes you may encounter an ?NF error for no reason at all. Not to worry. Just type in 'GOTO 1095' and if it still persists after 3 times, re-RUN the program.

Why does it do that? Well, the BASIC in the program is alright, its the side effects of the M/L that causes the bug.

Also, you can't hold me responsible for lost and/or damaged software by using this software.

The Listing:

```
0 GOTO5
1 '***** SECTOR BLASTER *****
2 '***** ALEX HARTMANN *****
3 SAVE"SECTBLAS:3":END
5 CLS:FOR A=&H1DA TO &H1F8:READ
1$ :V=VAL("&H"+1$):POKE A,V:NEXT
10 FORA=&H1FA TO &H20B:READ I:PO
KE A,I:NEXT:POKE&HEA,2:POKE&HEB,
DR:POKE&HEE,4:POKE&HEF,0
15 DATA7F,1,F9,8E,1,FA,A6,80,97,
ED,AD,9F,CO,4,96,FO,A7,88,11,BA,
1,F9,E7,1,F9,8C,2,C,26,E8,39
19 DATA1,12,5,16,9,2,13,6,17,10,
3,14,7,18,11,4,15,8,1,5,9,13,17,
3,7,11,15,12,16,2,6,10,14,18,4,8
20 CLEAR2000:CLS:PRINT9,"SECTOR
BLASTER"
21 PRINT64,"INPUT DRIVE NUMBER
==>"
22 AS=INKEY$:IFAS=""THEN22
23 IF AS<"0"OR AS>"3"THEN22ELSE
DR=VAL(AS)
24 POKE&HEB,DR
25 PRINT64," ==> PLACE DISK IN
DRIVE"DR"<=="..." PRESS <ENTE
R> TO START"," PRESS <SPACE>
TO ABORT"
30 AS=INKEY$:IF AS=""THEN30
31 IF AS="" THEN20
32 IF AS=CHR$(13)THEN40
33 GOTO30
40 '
1025 FOR TR=0TO34
1030 PRINT320,"TRACK: ";TR
1035 POKE&HEC,TR
1040 EXEC&H1DA
1045 IF PEEK(&H1F9)=0THEN1085
1050 FOR A=&H1FA TO &H20B
1055 SE=PEEK(A)
1060 E=PEEK(A+18)
1065 IF E=0THEN1080
1070 ER=ER+1
```

```
1075 TR(ER)=TR:SE(ER)=SE
1080 NEXTA
1085 NEXT TR
1095 CLS:PRINT9,"VERIFY DISK":P
RINTTAB(11)"RESULTS:":PRINT
1100 IF ER=0THENPRINT" YOUR
DISK CHECKS OK!":PRINT449," P
RESS <ANY KEY> TO RE-RUN";:EXEC4
4539:GOTO20
1105 IF ER=1THENPRINT" THER
E WAS 1 ERROR."ELSEPRINT" T
HERE WERE"ER"ERRORS."
1115 IF ER>8THEN1130
1120 PRINT:FORI=1TO ER:PRINT"
TRACK"TR(I)TAB(10)"SECTOR"SE
(I):NEXT
1130 PRINT449," PRESS <AN
Y KEY>";:EXEC44539:CLS
1135 CLS:PRINT9,"SECTOR BLASTER
":PRINT
1140 PRINT" NOW REMOVING TROUBLE
D SECTORS
1410 FORI=1TO ER:IF SE(I)>10THEN
SE(I)=10ELSE SE(I)=1:NEXTELSENE
XT
1415 GR=1:FORT=0TO34:FORS=1TO18S
TEP9
1420 FORI=1TO ER
1425 IF T=TR(I)AND S=SE(I)THEN G
(I)=GR
1430 NEXTI:GR=GR+1:NEXTS,T
1435 FORI=1TO ER:Q=G(I)
1440 DSKI$DR,17,2,AS,B$
1445 MID$(AS,Q,1)=CHR$(201)
1450 DSKO$DR,17,2,AS,B$
1455 NEXT
1460 DIR:PRINT"FREE = "FREE(DR)
1470 PRINT:PRINT" PRESS <ANY
KEY> TO RE-RUN"
1480 EXEC44539:GOTO20
```

HINT....

Merging two programs (Tape only)

If you have a tape system and you want to merge two BASIC programs together, try to follow:

- RENUMBER both programs so that they don't overlap.
- CLOAD the lower number program.
- Type in:

```
POKE25,PEEK(27) <ENTER>
POKE26,PEEK(26)-2 <ENTER>
d. CLOAD the second program
e. Type in:
POKE25,30 <ENTER>
POKE26,1 <ENTER>
If you want to continue
merging programs, go to step 'c'
and continue from there.
```

Keep programs out of sight,
but not out of memory.....

HIDE

32K ECB

by George McLintock

THIS IS A small two part utility which applies a feature of Basic described in the note, 'Some Aspects of Basic'. It can be used to prevent a part of a Basic program being listed or executed until some other function has been completed by the first part of the program. For example, to prevent a part of a program being listed or executed unless a password or code is known and entered correctly.

The utility can be used in association with COMPRESS to obtain a second level of code concealment. In use, the second part of the utility, which restores the full program, must be contained in that part of the program which is executed first.

The Listing:

```
0 GOTO10
3 SAVE"HIDEBAS:3":END
10 'DEMO OF HIDE UTILITY
    BY G.T.MCLINTOCK
15 'PART 2: RESTORE FULL PROGRAM
20 M=&H1DA:CK=0
30 FOR X= 0 TO 28
40 READ C$:Z=VAL("&H"+C$)
50 POKE M+X,Z:CK=CK+Z
60 NEXT X:READ C$
70 IF CK (<) VAL("&H"+C$) THEN PR
INT "ERROR IN LINE NO 200":STOP
100 EXEC M
200 DATA 9E,19,EC,84,27,04,AE,84
,20,F8,AF,84,9E,19,1F,12,30,04,A
6,80,26,FC,AF,A4,EC,84,26,F2,39,
D47
400 'LAST LINE SEEN
500 PRINT "NOW YOU SEE THE LOT A
GAIN"
510 STOP
600 'PART 1 - HIDE FROM LINE 500
610 FOR X=0 TO 29:READ C$:NEXT X
'PAST PART 1
620 M=&H1DA:CK=0
630 FOR X=0 TO 20
640 READ C$:Z=VAL("&H"+C$)
650 POKE M+X,Z:CK=CK+Z
660 NEXT X:READ C$
670 IF CK (<) VAL("&H"+C$) THEN P
RINT "ERROR IN LINE NUMBER 700":
STOP
680 EXEC M
690 STOP
700 DATA 9E,19,EC,02,27,0E,10,83
,01,F4,27,04,AE,84,20,F2,6F,80,6
F,80,39,7E8
```

PROGRAM LISTINGS

Program listings are provided in two formats. Assembler source code with comments to show the functions performed, and a Basic program with data statements to be poked into memory. The data statements contain checksums to help avoid typing errors.

The assembler source listings are ORG'ed at 32000 for Part 1, and 32100 for Part 2. The Basic program is set up in a structure that you might use. RUN 600 executes Part 1, and RUN executes Part 2. The ML code is held in DATA statements and POKE'd into the cassette buffer at Hex 1DA (equals M) for execution. The ML code is relocatable and can be executed from anywhere in memory. (Change

M for this)

Part 1 of the utility changes the address of the next line for line 500 to zeros, which effectively makes the previous line the new end of program. (See the note 'Some Aspects of Basic' for more detail). To change the line number used, poke the line number required into M+8 and M+9.

For example, if the line number before 600 was required, add the following after line 70

```
80 LN=600: H=INT(LN/256)
```

Part 2 of the utility restores the complete program. It removes the dummy end of the program marker set by program 1. Note that if Part 2 is executed out of sequence, garbage will be added to the end of the program.

ASSEMBLER LISTING

```
00010 * PART 1 - SET DUMMY END OF PROGRAM 7D00
00100 ORG 32000 7D00 9E 19
00110 START LDX <25 START OF BASIC PROGRAM 7D02 EC 02
00120 S0 LDD 2,X THIS LINE NUMBER 7D04 27 0E
00130 BEQ NOTF END OF PROGRAM 7D06 1083 01F4
00140 CMPD #500 CHECK FOR LINE 7D0A 27 04
00150 BEQ S1 GOT IT 7D0C AE 84
00160 LDX ,X NEXT LINE 7D0E 20 F2
00170 BRA S0 CONTINUE 7D10 6F 80
00180 S1 CLR ,X+ ZERO IT 7D12 6F 80
00190 CLR ,X+ 7D14 39
00200 NOTF RTS
00210 * PART 2 - RESTORES FULL PROGRAM 7D64
00220 ORG 32100 7D64 9E 19
00230 LDX <25 7D66 EC 84
00240 T0 LDD ,X 7D68 27 04
00250 BEQ T1 GOT DUMMY END 7D6A AE '84
00260 LDX ,X CONTINUE SEARCH 7D6C 20 F8
00270 BRA T0 7D6E AF 84
00280 T1 STX ,X MAKE IT NON ZERO
00290 * RESET ADDRESSES OF NEXT LINE 7D70 9E 19
00300 LDX <25 7D72 1F 12
00310 LINE TFR X,Y SAVE FOR LATER 7D74 30 04
00320 LEAX 4,X 7D76 A6 80
00330 LOOP LDA ,X+ FIND END OF LINE 7D78 26 FC
00340 BNE LOOP 7D7A AF A4
00350 STX ,Y STORE IN POINTER 7D7C EC 84
00360 LDD ,X TEST END OF PROGRAM 7D7E 26 F2
00370 BNE LINE 7D80 39
00380 RTS 7D00
00390 END START
00000 TOTAL ERRORS
```

Looking at CoCo from a different angle.....



TURTLE GRAPHICS

by Sean Murdoch

I AM SUBMITTING my first program to this magazine. The program is called Turtle Graphics and is written for the CoCo 3.

Turtle Graphics is a Graphics Utility that allows you to use complex angles and distances to draw scale drawings or pictures using the great resolution and colours of the CoCo 3.

The idea for this program came to me after using a similar program on the Apple at school. I immediately wanted the program for myself but had to wait another year before I received my CoCo 3. I found it hard to write the program until I learned trigonometry in Maths.

To use Turtle Graphics effectively you must read the instructions carefully and experiment with them I have learnt that the best way to learn computer programming is experimentation.

I hope to be submitting more programs in the near future but I am short of ideas.

The listing:

```

0 *****
1 *   TURTLE GRAPHICS   *
2 *****
3 *   INSTRUCTIONS     *
4 *****
5 *   PROGRAMMED BY:   *
6 *   SEAN MURDOCH    *
7 *   P.O. BOX 5      *
8 *   BRINGELLY.     *
9 *   N.S.W. 2171    *
10 *   (047) 74-8291  *
11 *****
12 POKE359,57:POKE65314,21
13 PALETTE12,63:PALETTE13,0
14 A$=STRING$(32,"*")
15 B$=STRING$(31,"*")
16 CLS
17 PRINTA$;
18 FORT=1TO14
19 PRINT"*"+STRING$(30,32)+"*";
20 NEXTT
21 PRINTB$;
22 POKE1535,106
23 PRINT@72,"TURTLE GRAPHICS";
24 PRINT@140,"FOR THE";
25 PRINT@205,"COCO3";
26 PRINT@264,"BY SEAN MURDOCH";

27 PRINT@329,"A PRODUCT OF:";
28 PRINT@393,"MERL SOFTWARE";
29 A1$="T4O2L4G03CEL8FEL4CO2L2GL
803EFEFL1GL2.EI4GB-B-L8GFL4GL2EL
8FEFL1.G"
30 A2$="O3L4EL8GEGGL4GGL2.AL4EL8
GEGGL4GAGL2EL4EL8GEGGL4GB-GGGL8G
FL1.E"
31 PLAYA1$+A2$
32 FORT=1TO100
33 PALETTE12,RND(63)
34 PALETTE13,RND(63)
35 NEXTT
36 PALETTERGB
37 CLS
38 PRINTTAB(8)"TURTLE GRAPHICS"
39 PRINTTAB(8)"*****"
40 PRINT:PRINT"TURTLE GRAPHICS I
S A GRAPHICS UTILITY THAT ALLO
WS YOU TO USE COMPLEX ANGLES AN
D DISTANCES TO DRAW HIGH RESOLUT
ION PICTURES ON THE GRAPHICS S
CREEN. THE FORMAT IS SIMILAR
TO LOGO IN WHICH YOU TELL TH
E COMPUTER TO
41 PRINT"GO RIGHT OR LEFT A SPEC
IFIC ANGLE AND TO GO FORWARD
S AT THAT ANGLE LEAVING A PATH OF
COLOURED PIXELS."
42 GOSUB121
43 CLS
44 PRINT"HERE ARE MANY DIFFEREN
T COMMANDS TO USE IN YOUR
QUEST TO DRAW DETAILED OR THREE
DIMENSIONAL DRAWINGS US
ING 'TURTLE GRAPHICS'. THESE
E WILL NOW BE EXPLAINED"
45 GOSUB121
46 CLS
47 PRINT"WOULD YOU LIKE THESE PU
T TO YOUR PRINTER FOR EASY R
EFERENCE?"
48 PRINT"(Y/N)"
49 I$=INKEY$
50 IF I$="Y" THEN END=-2 ELSE IF I$="N" T
HEN 51 ELSE 49
51 CLS
52 PRINT#D,"ENTERING COMMANDS
53 PRINT#D,"*****"
54 PRINT#D,"(<#> WILL DRAW THE PR
OGRAM"
55 PRINT#D,"(<Ex> WILL ALLOW YOU
TO CHANGE LINE NUMBER x
56 PRINT#D,"(<SAVE> WILL SAVE PRO
GRAM TO TAPE";
57 PRINT#D,"(<Ix> WILL ALLOW YOU
TO INSERT A LINE AT x
58 PRINT#D,"(<LOAD> WILL LOAD PRO
GRAM FROM TAPE"
59 PRINT#D,"(<*> WILL LIST PROGRA
M"

60 PRINT#D,"(<&> WILL LIST PROGRA
M TO PRINTER";
61 PRINT#D,"(<NEW> WILL CLEAR ALL
PROGRAMS AND START AGAIN"
62 GOSUB121:CLS
63 PRINT#D,"(<Mx-y> WILL COPY NUM
BERS x TO y ONTO THE END
64 PRINT#D,"(<Kx-y> WILL KILL NUM
BERS x TO y
65 PRINT#D,"(<$x> WILL COPY x ONT
O THE END
66 GOSUB121:CLS
67 PRINT#D,"DRAWING COMMANDS
68 PRINT#D,"*****"
69 PRINT#D,"(<Rx> MAKES TURTLE TU
RN RIGHT x DEGREES
70 PRINT#D,"(<Lx> MAKES TURTLE TU
RN LEFT x DEGREES"
71 PRINT#D,"(<Fx> MAKES TURTLE MO
VE FORWARD x PIXELS"
72 PRINT#D,"(<A> TURNS TURTLE ABO
UT 180 DEGREES
73 PRINT#D,"(<H> TAKES TURTLE BAC
K TO NORTH POSITION (FACING UPV
ARDS)
74 PRINT#D,"(<Nx> MOVES TURTLE FO
RWARD x PIXELS AND BACK TO T
HE POSITION IT STARTED FROM.
75 GOSUB121:CLS
76 PRINT#D,"(<U> MAKES THE TURTLE
'S PATH THE SAME AS THE BACKGRO
UND COLOUR
77 PRINT#D,"(<D> MAKES THE TURTLE
'S PATH THE SAME AS THE FOREGRO
UND COLOUR
78 PRINT#D,"(<Cx> SETS THE FOREGR
OUND COLOUR. (x CAN BE 0-15)
79 PRINT#D,"(<Bx> SETS THE BACKGR
OUND COLOUR TO x (0-15) AND CLEA
RS THE SCREEN AT THE BACKGR
OUND COLOUR
80 PRINT#D,"(<Px> PAINTS FROM THE
CURRENT TURTLE POSITION IN T
HE COLOUR x AND DOESN'T PAINT OV
ER CURRENT FOREGROUND COLOUR
81 GOSUB121:CLS
82 PRINT#D,"(<Gx> GOES TO NUMBER
x WHILE DRAWING. (SIMILAR TO
GOTO STATEMENT"
83 PRINT#D,"(<Xx,y> TAKES TURTLE
TO THE CO-ORDINATES OF x AN
D y BUT DOES NOT LEAVE A LINE"
84 GOSUB121:CLS
85 PRINT#D,"THE MOST POWERFUL CO
MMAND IS THEO(c,c,E)x
COMMAND. THIS COMMAN
D ALLOWS YOU TO PUT DIFFERENT FUN
CTIONS ALONGSIDE EACH OTHER
AND REPEAT THEM x TIMES. YOU CA
N HAVE A MAXIMUM OF 100 FUNCT
IONS (c)
86 PRINT#D,"SEPARATED BY COMMARS

```



```

AND ENDED WITH AN 'E'.
87 PRINT#D,"AN EXAMPLE IS A SQUA
RE. THIS CANBE DRAWN BY 'O(F40,R
90,E)4' WHICH WILL DRAW A S
QUARE WITH SIDES OF 40 PIXELS"
88 GOSUB121:CLS
89 PRINT#D,"WHILE THE TURTLE IS
DRAWING YOU MAY STOP ITS PROGRES
S BY PRESSING <S>"
90 PRINT#D,"ANOTHER USEFUL KEY I
S THE <BREAK> KEY. PRESS I
T TO GET OUTOF A SITUATION YOU D
ID NOT WANT"
91 PRINT#D,"IF YOU WANT A SCREEN
DUMP TO PRINTER THEN PRESS <
P> WHEN THE TURTLE HAS STOPPED D
RAWING."
92 GOSUB121:CLS
93 PRINT#D,"<'> CAN BE USED AS A
REM STATEMENT AND WILL I
GNORE THE REST OF THE LINE, AL
LOWING YOU TO MAKE REMARKS"
94 PRINT#D,"<QUIT> RESTORES THE
PALETTE AND TAKES YOU BACK TO BA
SIC COMMAND MODE"
95 GOSUB121:CLS
96 PRINT#D,"THE COMPUTER WILL PR
OMPT YOU WITH A LINE NUMBER.
YOU MAY ENTER YOUR COMMANDS
OR YOUR PROGRAM LINES. YOUR
ENTERING COMMANDS WILL NOT BE
STORED AS PART OF THE PROGRAM.
YOU MAY USEANGLES WITH DECIMAL
PLACES AND
97 PRINT#D,"AND DISTANCES WITH D
ECIMAL PLACES. YOUR ERRORS
WILL BE TRAPPED AND DECIPHER
ED.
98 GOSUB121:CLS
99 PRINT#D,"THE COLOURS ARE:"
100 PRINT#D,"0 - BLACK
101 PRINT#D,"1 - BUFF
102 PRINT#D,"2 - BLUE
103 PRINT#D,"3 - RED
104 PRINT#D,"4 - YELLOW
105 PRINT#D,"5 - CYAN
106 PRINT#D,"6 - MAGENTA
107 PRINT#D,"7 - ORANGE
108 PRINT#D,"8 - PURPLE
109 GOSUB121:CLS
110 PRINT#D,"9 - BROWN
111 PRINT#D,"10 - GREY
112 PRINT#D,"11 - DEEP BLUE
113 PRINT#D,"12 - BLACK
114 PRINT#D,"13 - GREEN
115 PRINT#D,"14 - GRASS-GREEN
116 PRINT#D,"15 - PURPLE-PINK
117 GOSUB121:CLS
118 PRINT"HAVE FUN WITH turtle g
raphics AND REMEMBER THAT THE
ONLY WAY TO LEARN TO USE IT IS
TO FOLLOW THE INSTRUCTIONS AND E
XPERIMENT WITH THEM".
119 POKE359,126
120 END
121 PRINTTAB(10)"<ANY KEY>";
122 IFINKEY$=""THEN122
123 RETURN
124 ' A PRODUCT OF MERL SOFTWARE
125 ' 28/12/86

```

```

0 '*****
1 '* TURTLE GRAPHICS *
2 '*****
3 '* BY *
4 '* SEAN MURDOCH *
5 '* P.O. BOX 5 *
6 '* BRINGELLY. *
7 '* N.S.W. 2171 *
8 '* (047) 74-8291 *
9 '*****
10 ' COCO3 VERSION
11 GOTO13
12 GOTO133
13 CLS
14 POKE65497,0
15 CLEAR10000
16 DIMA$(1000),Q$(100)
17 ON ERR GOTO12
18 ON BRK GOTO21
19 FORT=0TO15:READD1:PALETTET,D1
:NEXTT
20 DATA0,63,1,4,6,3,40,52,5,50,5
6,29,0,18,49,45
21 HSCREEN0:CH=160:CV=96:C1=CH:C
2=CV:F=1:B=0:AN=270:Z=0:Q=0
22 IFJ=0THENJ=1
23 PRINTJ");:LINEINPUTA$(J)
24 IFA$(J)=""THEN 23
25 A$=LEFT$(A$(J),1):B$=MID$(A$(
J),2,LEN(A$(J))-1)
26 IFA$="#"THENJ=J-1:GOTO40
27 IFA$="I"THEN75
28 IFA$="$"THENA$(J)=A$(VAL(B$))
29 IFA$="S"ANDB$="AVE"GOTO99
30 IFA$="L"ANDB$="OAD"THEN111
31 IFA$="*"GOTO123
32 IFA$="&"GOTO128
33 IFA$="N" AND B$="EW"THENCLS:R
UN
34 IFA$="Q"ANDB$="UIT"THENPOKE65
496,0:PALETTERRGB:END
35 IFA$="M"GOTO142
36 IFA$="K"GOTO142
37 IFA$="E"THEN179
38 J=J+1
39 GOTO21
40 HSCREEN2:HCOLORF,B:HCLS:FORW=
1TOJ
41 A$=LEFT$(A$(W),1):B$=MID$(A$(
W),2,LEN(A$(W))-1)
42 IFA$="R"THENAN=AN+VAL(B$):GOT
O70
43 IFA$="L"THENAN=AN-VAL(B$):GOT
O70
44 IFA$="A"THENAN=AN+180:GOTO70
45 IFA$="F"THENDI=VAL(B$):GOTO58
46 IFA$="H"THENAN=270:GOTO70
47 IFA$="N"THENDI=VAL(B$):Y=1:GO
TO58
48 IFA$="O"THEN83
49 IFA$="U"THENX3=1:GOTO70
50 IFA$="D"THENX3=0:GOTO70
51 IFA$="C"THENF=VAL(B$):HCOLORF
:GOTO70
52 IFA$="B"THENB=VAL(B$):HCLSB:G
OTO70
53 IFA$="P"THENHPAINT(C1,C2),VAL
(B$),F:GOTO70
54 IFA$="G"THENW=VAL(B$)-1:GOTO7
0
55 IFA$="X"THEN160
56 IFA$="'"THEN70
57 PRINT"ERROR IN"W:GOTO21
58 IFAN>360THENAN=AN-360
59 IFAN<0THENAN=AN+360
60 K=AN/57.29577951
61 H=COS(K)*DI

```

```

62 V=SIN(K)*DI
63 CH=CH+H:CV=CV+V
64 IFCH<0ORCH>320ORCV<0ORCV>191T
HEN133
65 IFX3=1THEN67
66 HLINE(FIX(C1+.5),FIX(C2+.5))-
(FIX(CH+.5),FIX(CV+.5)),PSET
67 IFINKEY$="S"THEN38
68 IFY=1THENY=0:CH=C1:CV=C2:GOTO
70
69 C1=CH:C2=CV
70 IFZ=1 THEN96
71 NEXTW
72 I$=INKEY$:IFI$=""THENPLAY"L25
5V501CB":GOTO72
73 IFI$="P"THEN165
74 GOTO38
75 CLS
76 IFB$=""THEN21
77 T=VAL(B$)
78 FORN=J TOT STEP-1
79 A$(N)=A$(N-1)
80 NEXTN
81 PRINTT");:LINEINPUTA$(T)
82 J=J+1:GOTO21
83 Z=1
84 E1=INSTR(1,A$(W),""):E2=INST
R(1,A$(W),")")
85 Q$=MID$(A$(W),E1+1,E2-E1-1)
86 Q=Q+1
87 I=INSTR(1,Q$,"")-1
88 Q$(Q)=MID$(Q$,1,I)
89 Q$=RIGHT$(Q$,LEN(Q$)-I-1)
90 IFLEFT$(Q$,1)="E"THEN92
91 GOTO86
92 FORG=1TOVAL(RIGHT$(A$(W),LEN(
A$(W))-E2))
93 FORO=1TOQ
94 A$=LEFT$(Q$(O),1):B$=RIGHT$(Q
$(O),LEN(Q$(O))-1)
95 GOTO42
96 NEXTO,G
97 Z=0:Q=0
98 GOTO71
99 '
100 CLS
101 J=J-1
102 INPUT"FILENAME";U$
103 POKE65496,0
104 OPEN"O",#-1,U$
105 FORT=1TOJ
106 PRINT#-1,A$(T)
107 NEXTT
108 CLOSE#-1
109 POKE65497,0
110 GOTO38
111 '
112 CLS
113 INPUT"FILENAME";U$
114 POKE65496,0
115 AUDI00N
116 J=J-1
117 OPEN" I",#-1,U$
118 J=J+1:INPUT#-1,A$(J)
119 IFEOF(-1)THEN120ELSE118
120 CLOSE#-1
121 POKE65497,0
122 GOTO38
123 CLS:FORT=1TOJ-1
124 PRINTT");A$(T)
125 IFT/10=INT(T/10)THENPRINT"PR
ESS ANY KEY TO CONTINUE";:EXEC44
539:CLS
126 NEXTT
127 GOTO21

```

continued on page 21



Opening a few more options with.....

COCO 3 UTILITIES

DISK UTILITY

by Fred Bisseling

THIS PROGRAM was public domain software for the CoCo 3. I have altered the CoCoutil/bas program to stop all the 'crashes' every time there was a typo or disk error. This was done by adding error trapping routines. I have also changed the screen driver routines but left in a couple of options for people who do not like mine.

The Listing:

```

0 GOTO10
1 ***** COCOUTIL *****
  **** FRED BISSELING ****
3 SAVE"140:3":END
10 'THIS PROGRAM MODIFIED BY
20 'RANDY POPPE FOR THE COCO 3-8
0 COLUMN
30 'ORIGINALLY WAS DIR80D BY
40 'DATALOG SOFTWARE FOR WORDPAK
50 'SET UP FOR GREEN SCREEN MONI
TOR
60 '
70 'MODIFIED BY FRED BISSELING-C
OOMA. THIS IS TO PROVIDE
80 'IMPROVED SCREEN DRIVER ROUTI
NES AND TO FIX 'BUGS' IN
90 'DISK ROUTINES.
100 'HAVE ALSO ADDED SOME ERROR
TRAPPING ROUTINES TO PREVENT
110 'THE PROGRAM CRASHING AT AWK
WARD MOMENTS DUE TO TYPO'S
120 '
130 'REMOVE OR SET 'REM'S' IN OR
DER TO SELECT YOUR DESIRED SCREE
N
140 'DRIVERS (line No.s 1320 to
1480) OR PRINTER BAUD RATE (line
150 'No.s 170 to 200): ERROR ROU
TINES (lines 1500 to 1730)
160 '
170 'SET PRINTER BAUD RATES
180 'POKE150 , 'SET AT 600
190 POKE150,41 'SET AT 1200
200 'POKE150, 'SET AT 2400
210 '
220 GOSUB 1320
230 CLEAR6000,&H7FEE:VERIFYON
240 ON ERR GOTO 1510
250 F1$=STRING$(79,"=")
260 DIMP$(55,4):FT$(0)="B":FT$(1)
)="D":FT$(2)="M":FT$(3)="S"
270 CLS
280 PRINT:PRINTTAB(22);"C O C O
- 3 Disk Utility"
290 PRINT:PRINTF1$:PRINT"ENTER D
RIVE #";
300 N$=INKEY$:IFN$=""THEN300ELSE
IFN$=CHR$(13)THENN$="0"
310 IFVAL(N$)<ORVAL(N$)>4THEN30
0
320 N=VAL(N$):ATTR3,0,B:PRINT#;T
AB(46);FREE(N);"GRANS FREE":PRIN
T
330 ATTR3,0
340 R=0
350 FORZ=3TO11
360 DSKI$N,17,Z,A$,B$
370 X$=A$:GOSUB410
380 X$=B$:GOSUB410
390 NEXTZ
400 GOTO510
410 FORB=1TO128STEP32
420 R=R+1
430 P$(R,1)=MID$(X$,B,8)
440 IFLEFT$(P$(R,1),1)=CHR$(255)
THENR=R-1:GOTO510
450 IFLEFT$(P$(R,1),1)=CHR$(0)TH
ENR=R-1:GOTO490
460 P$(R,2)=MID$(X$,B+8,3)
470 P$(R,3)=MID$(X$,B+11,1)
480 P$(R,4)=MID$(X$,B+12,1)
490 NEXTB
500 RETURN
510 RESTORE
520 FORK=1TO R
530 READ TB:IF TB=99 THEN RESTOR
E:PRINT#DE,CHR$(13);:GOTO530
540 PRINT#DE,TAB(TB);:PRINT#DE,U
SING"###";K;:PRINT#DE,CHR$(32);P$
(K,1);"/";P$(K,2); " ";:PRINT#DE,
FT$(ASC(P$(K,3)));:IFASC(P$(K,4)
)=255THENPRINT#DE,"";
550 IFK=R THENPRINT#DE
560 NEXTK:RESTORE
570 PRINT#DE:PRINT#DE,F1$:IFDE=-
2THENDE=0:GOTO510
580 ATTR3,0:PRINTTAB(9);"[A]scii
[L]oad,R [K]ill [R]ename [C
]opy [P]rint [T]op";" ";
590 ATTR3,0
600 AK$=INKEY$:IFAK$=""THEN600
610 IFAK$="L"THEN740
620 IFAK$="K"THEN800
630 IFAK$="R"THEN890
640 IFAK$="C"THEN920
650 IFAK$="T"THENRUN
660 IFAK$="A"THEN1050
670 IFAK$="P"THEN1020ELSE600
680 '
690 DE=0:PRINT:PRINTF1$:PRINT"Dr
ive";N;" Free grans=";FREE(N
):PRINT:RETURN
700 '
710 PRINT:PRINT"Selected=";AK$:L
INEINPUT"ENTER FILE NUMBER:";FI
S:F1=VAL(F1$):IFF1=0THEN510
720 PROG$=P$(F1,1)+"/"+P$(F1,2):
RETURN
730 '
740 GOSUB710:IF ASC(P$(F1,3))<0
THENPRINT"NOT A BASIC PROGRAM":F
ORX=1TO1500:GOTO510 ELSE PRINT"L
OADING "PROG$
750 PR$=PROG$+":"+N$:FORX=0TO13:
POKE&H7FEF+X,ASC(MID$(PR$,X+1,1)
):NEXT
760 FILES2,256:CLEAR200,&H7FEE '
RESET BASIC RAM
770 FORX=0TO13:PR$=PR$+CHR$(PEEK
(&H7FEF+X)):NEXT
780 LOAD PR$,R:END
790 '
800 P=1:GOSUB1280:GOSUB710
810 GOSUB1220
820 PRINT"OK TO KILL "PROG$"..
.... Y or N?";
830 K$=INKEY$:IFK$=""THEN830
840 IFK$="N"THEN510ELSEIFK$="Y"
HEN860
850 GOTO830
860 PRINT "KILLING "PROG$:KILL
PROG$+":"+N$:IF FL THEN 870 ELSE
810
870 GOSUB690:GOTO340
880 '
890 GOSUB710:LINEINPUT"NEW FILE
NAME (XXXXXXXX/XXX) ? ";N$
900 RENAME PROG$+":"+N$ TO N$+
":"+N$:GOSUB690:GOTO340
910 '
920 P=1:GOSUB1280:GOSUB710
930 PRINT"SOURCE DRIVE #N":INPU
T"DESTINATION DRIVE #";DD$
940 IFDD$=""THEN510
950 GOSUB1220
960 PRINT"Copying... ";PROG$
970 IFDD$=N$ THEN COPY PROG$:GOT
O990
980 COPY PROG$+":"+N$ TO PROG$+
":"+DD$
990 IF FL THEN 1000 ELSE 950
1000 RUN
1010 '
1020 ON ERR GOTO 1510:PRINT:PRIN
T"PRINTING DIRECTORY";:DE=-2:PRI
NT#DE,TAB(26);"D I S K D I R E
C T O R Y":PRINT#DE,F1$:PRINT#D
E:GOTO520
1030 DATA 0,3,3,3,99
1040 '
1050 GOSUB710
1060 IFASC(P$(F1,4))<255THENPRI
NT:PRINT"Not an ascii file....":
FORX=1TO2000:NEXT:GOTO510
1070 INPUT"<S>screen or <P>printer
";AK$:IFAK$="P"THENDE=-2
1080 OPEN"D",#1,PROG$+":"+N$,1
1090 FIELD#1,1ASH$:BY=LOF(1)
1100 CLS:PRINT"File: ";PROG$;"
continued on page 39

```

SUPER INTELLIGENT SCREEN DUMP

32K ECB + Super80 or Amust printer

by Jeff Larsen

THIS SCREEN DUMP will keep you in the picture. The program will dump one or two PMODE 4 graphics pages to any printer compatible with the Super80 or Amust series. These printers all have an 8 bit image mode of 1280 dots/line, which is started by control codes <ESC>+L.

This program will stand alone ...

```
CLEAR 200,&H74FF:LOADM"SUPER"  
:EXEC
```

... or it can be interfaced with a basic program. Super is written in position independent code and can be offset loaded to any area of memory, provided an appropriate clear has been done first.

Super is really four dumps in one. The first one gives a pixel for dot dump, which results in a very small picture. This picture is available from Super menu selection 1.

The second dump uses half a page on your printer to dump one pmode 4 graphics page.

This dump has the option of expanding the picture four or five times across the page.

The fourth dump is an extra large dump.

It prints the picture sideways and fills a sheet of paper with one graphics page.

The first three dumps all have the option of printing one or two PMODE 4 graphics pages. This caters for CoCoMax pictures that

use both hires pages. With all the dumps, it is also possible to just print the PMODE 4,2 page. As the program gets its starting point from a basic variable, all you need to do is type PMODE4,2 <ENTER> before executing super.

The next feature of these dumps is the ability to invert the picture before printing.

This will let you print a negative image. This inversion is also important if you are to avoid damaging your printer head.

Buried in the back of my printer manual is a warning on printer head temperature protection. This recommends that continuous usage of one pin be kept within one line of 640 dots, or a stop time of double the printing time be provided after one line of printing.

Super dump does NOT provide any stop times. The only protection is a simple pixel counting routine which will warn you if your picture contains large amounts of black. If you get this warning before printing, I suggest you either answer yes to the question "invert the picture" or during the printing operate the on line key to give the printer head time to cool.

Printer heads are expensive!

This warning should not stop you from using super. All machine language screen dumps are liable to overheat your printer head. The others I have

seen do not check areas of black, or offer any warning.

Super can also be interfaced with a basic program. When this interface is used with a USR call, super does not show any menu selections and all options must be chosen by the value passed by the basic program. This is fully explained in the source code for super.

A further illustration of how to call super from basic is given in the programme "Panorama" (by Dave Allen), which uses super to print two CoCoMax pictures across the page.

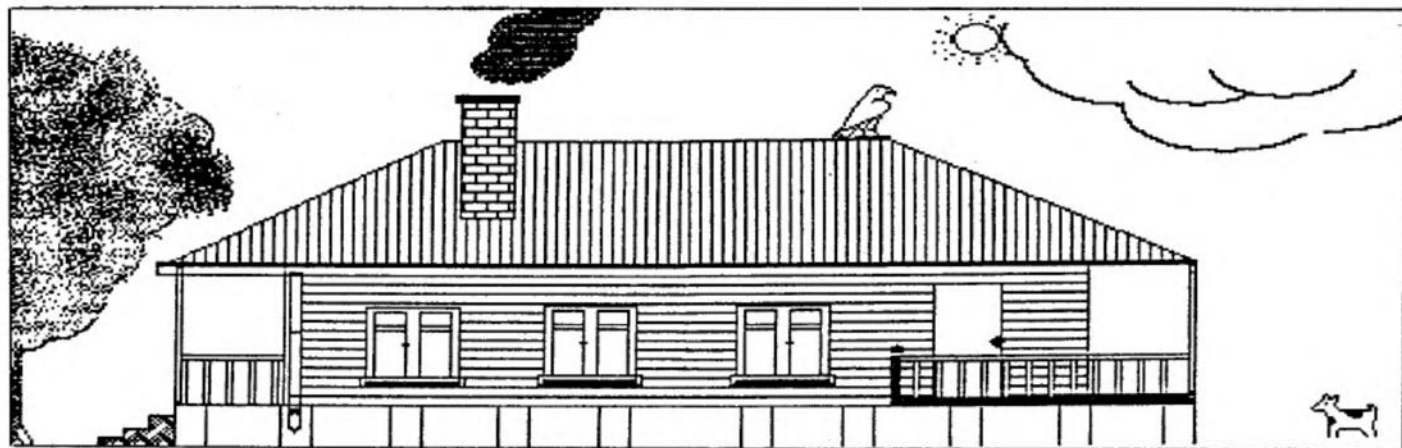
This allows an extra wide picture, drawn in two halves to be printed as one picture.

Super also displays the picture while printing. Super shows the hires screen and inverts each byte as it is printed. This results in a display which shows how far the program has progressed in its printing. Of course the picture is restored to its original form before super finishes.

The first three dumps have a feature classed as white skip. Before the printer is set to graphics mode, the next line is examined. If this line is blank a carriage return is sent to the printer and the program proceeds to the next line.

This will speed up the printing of a picture with large areas of white.

□



```

00100 $SUPER HI-RES SCREEN DUMP
00110 $1/10/86
00120 $BY JEFF LARSEN
00130 $43 YANGOORA ST.
00140 $CAIRNS QLD 4871
00150 $
00160 $WITH HELPFUL IDEAS FROM
00170 $DAVE ALLEN
00180 $
00190 $
00200 $THIS PROGRAMME WILL STAND
00210 $ALONE, OR CAN BE CALLED
00220 $FROM BASIC WITH A=USR0(X)
00230 $OPTIONS ARE:
00240 $SIZE OF DUMP 1. SMALL
00250 $                2. MEDIUM
00260 $                3. LARGE
00270 $NONE OR TWO PAGES
00280 $SCREEN INVERTED
00290 $THE MEDIUM DUMP CAN ALSO
00300 $HAVE THE OPTION OF 4X OR
00310 $5X ACROSS MAGNIFICATION
00320 $
00330 $IF THIS IS CALLED USING
00340 $A USR CALL THEN THE VALUE
00350 $PASSED SHOULD BE:
00360 $1 FOR A SMALL DUMP
00370 $2 FOR A MEDIUM DUMP WITH
00380 $4X ACROSS THE PAGE
00390 $3 FOR A MEDIUM DUMP WITH
00400 $5X ACROSS THE PAGE
00410 $4 FOR LARGE DUMP
00420 $ADD 16 TO ABOVE VALUE TO
00430 $DO 2 PNODE 4 SCREENS
00440 $AND 256 TO NOT INVERT SCREEN
00450 $EG: DEFUSR0=WH7500
00460 $ A=USR0(3*16+256)
00470 $FOR A MEDIUM DUMP WITH 5X
00480 $ACROSS THE PAGE,
00490 $2 GRAPHICS PAGES DUMPED,
00500 $AND NOT INVERTED SCREEN
00510 $
00520 $
00530 INTCNV EQU 003E0
00540 DEVNUM EQU 06F
00550 POLCAT EQU 0A000
00560 CHR0UT EQU 0A002
00570 $
00580 ORG 07500 HIGH MEMORY
00590 START PSHS A,B,X,Y,CC SAVE REGISTERS
00600 CLR 0FF40 DRIVES OFF
00610 JSR INTCNV GET PASSED VALUE
00620 TSTB HAS A VALUE BEEN PASSED?
00630 LBEQ ASK NO GO ASK WHAT SORT OF DUMP
00640 START1 STA INVERT,PCR SAVE INVERT VALUE
00650 STB CHDST,PCR SAVE WHICH DUMP
00660 LDX 0BA START OF GRAPHICS PAGES
00670 LEAX 01000,X END OF FIRST PAGE
00680 STX MID,PCR SAVE IT
00690 CNPB 000F ARE WE DOING TWO PAGES?
00700 BLO 0NEP NO, ONLY ONE PAGE
00710 LEAX 01000,X END OF SECOND PAGE
00720 0NEP STX ENDO,PCR SAVE IT
00730 ANDB 07 MASK OUT PAGE DATA
00740 CNPB 04 ERROR TRAPPING
00750 BHI EXIT GO IF WRONG
00760 LBSR CPRT CHECK PRINTER ON
00770 BNE EXIT RETURN IF ASKED TO
00780 TST INVERT,PCR INVERT SCREEN?
00790 DER NOINV NO
00800 LBSR INV YES
00810 NOINV CLR EXPAND,PCR SETUP EXPANSION FOR 2 AND 3
00820 CNPB 03 EXPANDING?

```

```

00830 BNE NOEX NO
00840 INC EXPAND,PCR YES
00850 NOEX LDX 0BA START OF GRAPHICS PAGE
00860 LDY 00 Y HOLDS COUNT OF SET PIXELS
00870 ABYTE CMPX ENDO,PCR HAVE WE FINISHED CHECKING?
00880 BEQ BCONT YES, GO DO DUMP
00890 LDA ,X GET A BYTE
00900 BEQ ABYTE NO SET PIXELS
00910 CPIX TSTA
00920 BEQ ABYTE IF NONE SET, GO DO MORE
00930 ASLA SHIFT ONE PIXEL INTO CARRY BIT
00940 BCC CPIX NOT SET, GO LOOK AT ANOTHER
00950 LEAY I,Y GOT A SET ONE, INCREMENT COUNTER
00960 CMPY 00000 REACHED LIMIT
00970 BLO CPIX IF NOT, CHECK SOME MORE
00980 CNPX MID,PCR REACHED LIMIT AT LESS THAN ONE PAGE?
00990 BLO WARNU GO WARN USER
01000 CNPY 00 DOING MORE THAN ONE PAGE
01010 BNE CPIX CHECK IF PIXEL COUNT IS MORE THAN 0FFFF
01020 WARNU LBSR CLS CLEAR SCREEN
01030 LEAX WARN,PCR POINT TO WARNING MESSAGE
01040 LBSR PRINT PUT IT ON THE SCREEN
01050 WLOOP JSR [POLCAT] GET A KEYPRESS
01060 BEQ WLOOP LOOP UNTIL PRESSED
01070 CNPA 0'Y HAS USER PRESSED 'Y'?
01080 BEQ BCONT YES, CONTINUE WITH DUMP
01090 BRA EXIT OTHERWISE BACK TO BASIC
01100 BCONT LBSR DISPLY SHOW HIRES SCREEN
01110 DEC DEVNUM PUT -2 IN DEVNUM
01120 DEC DEVNUM SO OUTPUT GOES TO PRINTER
01130 LDB CHDST,PCR GET USER'S INPUT
01140 ANDB 07 MASK OUT UNWANTED INFORMATION
01150 CNPB 01 WHICH DUMP ARE WE DOING?
01160 LBEQ DUMP1
01170 CNPB 04
01180 LBEQ DUMP3
01190 LBRA DUMP2
01200 EXIT CLR DEVNUM RESET OUTPUT TO SCREEN
01210 PULS A,B,X,Y,CC,PC REGAIN REGISTERS AND RTS
01220 ASK LBSR CLS CLEAR THE SCREEN
01230 CLR DEVNUM OUTPUT TO SCREEN
01240 LEAX TITLE,PCR POINT AT FIRST STRING
01250 BSR PRINT PUT IT ON THE SCREEN
01260 LEAX QUEST1,PCR POINT AT FIRST QUESTION
01270 BSR PRINT PUT IT ON THE SCREEN
01280 AGN1 BSR KEYPR GET A KEYPRESS
01290 CNPA 0'1 IS IT '1'?
01300 BLO AGN1 TOO LOW, IGNORE IT
01310 CNPA 0'4 IS IT A '4'?
01320 BHI AGN1 TOO HIGH, IGNORE IT
01330 PSHS A SAVE GOOD RESPONSE
01340 JSR [CHR0UT] SHOW RESPONSE ON SCREEN
01350 LDA 000D GO TO NEXT LINE
01360 JSR [CHR0UT]
01370 PULS A REGAIN KEYPRESS
01380 SUBA 0'0 MAKE IT DECIMAL
01390 STA CHDST,PCR SAVE IT
01400 CNPA 04 DUMP NO 4?
01410 BEQ SKIP YES, SKIP NEXT QUESTION
01420 LEAX QUEST2,PCR POINT AT QUESTION STRING
01430 BSR PRINT PRINT IT
01440 AGN2 BSR KEYPR GET A KEYPRESS
01450 CNPA 0'1 IS IT A '1'?
01460 BLO AGN2 TOO LOW, TRY AGAIN
01470 CNPA 0'2 IS IT A '2'?
01480 BHI AGN2 TOO HIGH, TRY AGAIN
01490 PSHS A SAVE IT
01500 JSR [CHR0UT] PRINT IT ON SCREEN
01510 LDA 000D GO TO NEXT LINE
01520 JSR [CHR0UT]
01530 PULS A REGAIN KEYPRESS
01540 SUBA 0'0+1 MAKE IT A BINARY VALUE (0 OR 1)
01550 LDB 016 MULTIPLY BY 16

```

#1560	MUL		SHIFT VALUE IN A	#2290	BEQ	F	YES, FINISHED	
#1570	ADDB	CMDST,PCR		#2300	JSR	{CHROUT}	NO, PRINT IT	
#1580	STB	CMDST,PCR	SAVE IT WITH OTHER RESPONSE	#2310	BRA	PRTD	GO FOR ANOTHER	
#1590	SKIP	LEAX	QUEST3,PCR POINT TO NEXT QUESTION	#2320	F	RTS	FINISHED	
#1600	BSR		PRINT PRINT IT	#2330	{START OF DUMP 1			
#1610	AGN3	BSR	KEYPR GET A KEYPRESS	#2340	DUMP1	LEAX	OPSET,PCR POINT TO PRINTER CONTROL CODES	
#1620	LDB	CMDST,PCR	GET SAVED ANSWERS, SO FAR	#2350	BSR	PRTD	SEND TO PRINTER	
#1630	CMPA	'Y	IS KEYPRESS A 'Y'?	#2360	LDX	#BA	POINT TO START OF GRAPHICS PAGES	
#1640	BNE	NO		#2370	LEAX	31,X	ADJUST POINTER BEFORE ENTERING LOOP	
#1650	JSR	{CHROUT}	SHOW IT ON SCREEN	#2380	NLINE	LEAX	-31,X ADJUST POINTER A LINE, LESS ONE	
#1660	CLRA		SET VALUE OF A FOR INVERSION OF	#2390	NLINE1	PSHS	X SAVE POINTER	
#1670	LBRA	START1	GO DO DUMP GRAPHICS SCREEN	#2400	CMPX	ENDG,PCR	ARE WE FINISHED?	
#1680	NO	CMPA	'M' WAS 'M' PRESSED?	#2410	BEQ	FIN	YES, GO	
#1690	BNE	AGN3	GO TRY AGAIN FOR A GOOD KEYPRESS	#2420	CMPX	NID,PCR	DONE ONE PAGE?	
#1700	JSR	{CHROUT}	SHOW IT	#2430	BLO	NOTHD	NO, GO DO MORE	
#1710	LDA	#1	SET VALUE OF A FOR NO INVERSION	#2440	LDA	NID,PCR	MSB OF SECOND PHODE 4 PAGE	
#1720	LBRA	START1	GO DO DUMP	#2450	BSR	DISP2	DISPLAY IT	
#1730	{SUBROUTINES			#2460	LBY	ENDG,PCR	GET END OF DUMP	
#1740	PRT2	JSR	{CHROUT} PRINT A CHARACTER	#2470	STY	NID,PCR	STORE IN NID, SO WE DON'T DO THIS AGAIN	
#1750	PRINT	LDA	,X+ GET A CHARACTER	#2480	NOTHD	LDX	,S REGAIN POINTER TO CURRENT LINE, LEAVING IT ON THE STACK	
#1760	BNE	PRT2	NOT #, PRINT IT					
#1770	RTS		FINISHED	#2490	CLRA		255 TO TEST	
#1780	{GET A KEYPRESS			#2500	STLOOP	TST	,X+ TEST FOR A BLANK LINE	
#1790	KEYPR	JSR	{POLCAT} USE ROM ROUTINE	#2510	BNE	CONT1	NOT BLANK, GO DO SOME PRINTING	
#1800	BEQ	KEYPR	NO KEY PRESSED, TRY AGAIN	#2520	DECA		KEEP COUNT	
#1810	RTS		GOT KEYPRESS, RETURN	#2530	BNE	STLOOP	FINISHED TESTING?	
#1820	{INVERT GRAPHICS PAGES			#2540	CLRA		255 TO DO	
#1830	INV	LDX	ENDG,PCR POINT TO END OF GRAPHICS SCREENS	#2550	PULS	X	POINT TO BEGINNING OF CURRENT LINE	
#1840	INV2	COM	,X INVERT A BYTE	#2560	CONLOP	COM	,X+ SHOW USER WHAT WE ARE UP TO	
#1850	CMPX	#BA	REACHED THE START OF GRAPHICS SCREENS?	#2570	DECA		KEEP COUNT	
#1860	BHI	INV2	NO, GO DO MORE	#2580	BNE	CONLOP	GO DO MORE	
#1870	RTS		FINISHED	#2590	LDA	#00	LINE IS BLANK	
#1880	{CHECK IF PRINTER ON LINE			#2600	JSR	{CHROUT}	SKIP IT	
#1890	CPRT	LDA	#FF22	#2610	BRA	NLINE1	GO DO NEXT LINE	
#1900	ANDA	#1	GET PRINTER READY BIT	#2620	CONT1	PULS	X RESTORE POINTER TO CURRENT LINE	
#1910	BNE	POFF	IF LOW, GO TELL USER	#2630	JSR	OPRT0	SET GRAPHICS MODE ON PRINTER	
#1920	RTS		OK, PRINTER ON	#2640	LDB	#32	NUMBER OF LINES DOWN	
#1930	POFF	BSR	CLS CLEAR THE SCREEN	#2650	STB	LCOUNT,PCR	SAVE IT	
#1940	CLR	DEVNUM	OUTPUT TO SCREEN	#2660	NBYTE	LDB	#0	NUMBER OF BYTES DOWN
#1950	LEAX	ERRMS0,PCR	POINT TO MESSAGE	#2670	STB	BCOUNT,PCR	SAVE IT	
#1960	BSR		PRINT SHOW IT ON SCREEN	#2680	LEAY	STORE,PCR	POINT Y AT TEMP STORAGE	
#1970	BSR	KEYPR	GET A KEYPRESS	#2690	MLOOP	LDA	,X GET A BYTE	
#1980	CMPA	'Y	IS IT A 'Y'?	#2700	STA	,Y+	SAVE IT	
#1990	BEQ	BASIC	YES, GO	#2710	COM	,X	SHOW USER	
#2000	BRA	CPRT	NO, CHECK PRINTER AGAIN	#2720	LEAX	32,X	POINT AT NEXT BYTE	
#2010	BASIC	LDB	#1 SET FLAG FOR A RETURN	#2730	DECB		DECREMENT COUNTER	
#2020	RTS			#2740	BNE	MLOOP	GO DO ANOTHER	
#2030	CLS	LDX	#9400 POINT TO START OF TEXT SCREEN	#2750	BLOOP	LEAY	STORE,PCR POINT TO BEGINNING OF TEMP STORE	
#2040	STX	#00	RESTORE CURSOR POSITION	#2760	LDB	#0	SET COUNTER	
#2050	LDB	#0600	TWO SPACE CHARACTERS	#2770	SLOOP	ASL	,Y+ GET A PIXEL	
#2060	CLSLOP	STD	,X++ STORE IN VIDEO RAM	#2780	ROLA		PUT INTO ACC A	
#2070	CMPX	#9600	END OF TEXT SCREEN?	#2790	DECB		KEEP COUNT	
#2080	BNE	CLSLOP	LOOP IF NOT FINISHED	#2800	BNE	SLOOP	DO ANOTHER	
#2090	RTS			#2810	JSR	{CHROUT}	SEND TO PRINTER	
#2100	DISPLY	LDA	#F8 SHOW PHODE 4 PAGE	#2820	DEC	BCOUNT,PCR	DONE ALL BITS IN THIS BLOCK?	
#2110	STA	#FF22		#2830	BNE	BLOOP	GO DO ANOTHER	
#2120	STA	#FFC3	SET V00	#2840	DEC	LCOUNT,PCR	LINE COUNTER	
#2130	STA	#FFC5		#2850	BEQ	NLINE	GO DO NEXT LINE	
#2140	LDA	#BA	MSB OF START OF GRAPHICS PAGES	#2860	LEAX	-255,X	POINT AT BEGINNING OF NEXT BLOCK	
#2150	DISP2	LSRA		#2870	BRA	NBYTE	GO DO MORE	
#2160	LDB	#07	COUNTER	#2880	FIN	LEAS	2,S CLEAN UP STACK	
#2170	LDX	#FFC6	POINT TO SAM REGISTERS	#2890	LEAX	RES0AT,PCR	POINT AT PRINTER RESET CODES	
#2180	DISLOP	RORA	CHECK A BIT	#2900	LBSR	PRTD	SEND TO PRINTER	
#2190	BHS	DIS1	WAS BIT SET?	#2910	TST	INVERT,PCR	IS SCREEN INVERTED?	
#2200	STA	1,X	RESET SAM REGISTER	#2920	BNE	GO	NO, FINISHED	
#2210	BRA	DIS2		#2930	LBSR	INV	PUT GRAPHICS MEMORY BACK TO ORIGINAL	
#2220	DIS1	STA	,X	#2940	GO	LBRA	EXIT BACK TO BASIC	
#2230	DIS2	LEAX	2,X POINT TO NEXT SAM REGISTER	#2950	OPRT0	PSHS	X SAVE X	
#2240	DECB		DECREMENT COUNTER	#2960	LEAX	OP#,PCR	POINT TO PRINTER CONTROL CODES	
#2250	BNE	DISLOP	DONE ??	#2970	LBSR	PRTD	SEND TO PRINTER	
#2260	RTS		FINISHED	#2980	PULS	X,PC	RESTORE X AND RTS	
#2270	PRTD	LDA	,X+ GET DATA	#2990	{START OF DUMP 2			
#2280	CMPA	#0FF	LAST ONE?					

#3000	DUMP2	LEAX	PRTD,PCR	POINT TO PRINTER CONTROL CODES	#3700	SPRT2	TST	EXPAND,PCR	ARE WE DOING 4X ACROSS?				
#3010	LBSR	PRTD	SEND TO PRINTER	#3710	BEQ	LSKIP	GO TO CORRECT PRINTER CODES	#3720	LEAX	TWOD1,PCR	POINT AT 5X ACROSS PRINTER CODES		
#3020	LDB	#BA	POINT TO START OF GRAPHICS PAGES	#3730	LDBA	PRTD	SEND TO PRINTER	#3740	LEAX	TWOD2,PCR	POINT AT 4X PRINTER CODES		
#3030	NXTL	LEAX	#20,X	POINT TO NEXT LINE DOWN	#3750	LDBA	PRTD	SEND TO PRINTER	#3760	START OF DUMP 3			
#3040	LDA	#-#20	NUMBER OF BYTES ACROSS PAGE	#3770	DUMP3	LEAX	PRTD,PCR	POINT TO PRINTER CODES	#3780	LBSR	PRTD	SEND TO PRINTER	
#3050	STA	LCOUNT,PCR	SAVE IT	#3790	LDB	#BA	POINT TO START OF GRAPHICS PAGES	#3800	LEAY	#17E#,X	POINT Y TO LAST LINE OF GRAPHICS PAGES		
#3060	PSHS	X	SAVE POINTER TO GRAPHICS	#3810	CLRB	ACC	#	HOLDS OFFSET ACROSS THE PAGE	#3820	LOP3	CLR	FLAG,PCR	PIXEL OFFSET
#3070	TESTL	TST	-#20,X	IS THIS BYTE BLANK?	#3830	LOP2	LBSR	SPRT	SET PRINTER TO GRAPHICS MODE	#3840	TFR	Y,U	SAVE Y
#3080	BNE	CONT3	NO, GO DO SOME PRINTING	#3850	LOOP	PSHS	#	SAVE OFFSET	#3860	LDA	#,Y	GET A BYTE FROM GRAPHICS PAGE	
#3090	TST	,X+	IS THIS ONE BLANK?	#3870	LDB	FLAG,PCR	GET PIXEL OFFSET	#3880	BEQ	FIRST	IF FIRST TIME THROUGH WE WANT BITS 6 AND 7		
#3100	BNE	CONT3	NO, GO PRINT IT	#3890	CMPB	#1		#3900	BEQ	SECOND	GET BITS 4 AND 5		
#3110	INCA		KEEP COUNT	#3910	CMPB	#2		#3920	BEQ	THIRD	GET BITS 2 AND 3		
#3120	BNE	TESTL	TEST MORE, UNTIL FINISHED LINE	#3930	BRA	FOURTH	GET BITS # AND 1	#3940	FIRST	LSRA	SHIFT UNTIL PIXELS		
#3130	PULS	X	RESTORE POINTER	#3950	LSRA		WE WANT ARE IN THE	#3960	SECOND	LSRA	LOWER 2 BITS OF ACC A		
#3140	LDA	#-#20	NUMBER OF BYTES ACROSS PAGE	#3970	LSRA			#3980	THIRD	LSRA			
#3150	COMPL	COM	-#20,X	SHOW USER WHAT WE ARE UP TO	#3990	LSRA		#4000	FOURTH	ANDA	#3	MASK OUT ANY OTHER PIXELS LEFT	
#3160	COM	,X+	DOING THIS ROW OF PIXELS ALSO	#4010	LEAX	EXTAB,PCR	POINT TO EXPANSION TABLE	#4020	LDA	A,X	EXPAND TWO PIXELS INTO 6 DOTS FOR PRINTING		
#3170	INCA		KEEP COUNT	#4030	JSR	[CHROUT]	PRINT IT THREE TIMES	#4040	JSR	[CHROUT]			
#3180	BNE	COMPL	DO MORE IF NOT AT END OF LINE	#4050	JSR	[CHROUT]		#4060	PULS	#	REGAIN OFFSET ACROSS THE PAGE		
#3190	LDA	#000	LAST LINE WAS BLANK	#4070	LEAY	-#20,Y	POINT AT NEXT LINE UP	#4080	CMPLY	#BA	DONE A PAGE?		
#3200	JSR	[CHROUT]	TELL PRINTER TO SKIP IT	#4090	BPL	LOOP	DO MORE IF PAGE NOT FINISHED	#4100	LDA	#000	START ON NEW LINE ON PRINTER		
#3210	BRA	NXTLY	DO NEXT LINE	#4110	JSR	[CHROUT]	SEND IT TO PRINTER	#4120	TFR	U,Y	RESTORE Y		
#3220	CONT3	LBSR	SPRT2	SET PRINTER TO GRAPHICS MODE	#4130	INC	FLAG,PCR	NEXT PAIR OF PIXELS	#4140	LDA	FLAG,PCR	CHECK IF A BYTE HAS BEEN DONE	
#3230	PULS	X	GET POINTER	#4150	CMPLY	#4		#4160	BEQ	NEXTL	GO DO NEXT ROW OF BYTES		
#3240	NXTB	LDA	#-#	# PIXELS PER BYTE	#4170	BRA	LOP2	THIS ROW NOT FINISHED	#4180	BRA	LOP2	THIS ROW NOT FINISHED	
#3250	STA	BCOUNT,PCR	SAVE IT	#4190	INCB		NEXT ROW	#4200	CMPB	##20	DONE 32 ACROSS?		
#3260	LDA	-#20,X	GET FIRST BYTE	#4210	BNE	LOP3	DO NEXT ROW IF NOT	#4220	LEAX	RESDAT,PCR	POINT TO PRINTER RESET CODES		
#3270	STA	STORE,PCR	SAVE IT	#4230	LBSR	PRTD	SEND TO PRINTER	#4240	TST	INVERT,PCR	ARE GRAPHICS PAGES INVERTED?		
#3280	LDB	,X	GET SECOND BYTE	#4250	BNE	FFIN	IF NOT FINISHED	#4260	LBSR	INV	PUT GRAPHICS PAGES BACK HOW THEY STARTED		
#3290	AGAIN	LDA	STORE,PCR	GET FIRST BYTE	#4270	FFIN	EXIT AND RETURN	#4280	SPRT	LEAX	PGRAP3,PCR	POINT TO PRINTER CODES	
#3300	ROLA		GET A PIXEL	#4290	LDBA	PRTD	SEND TO PRINTER	#4300	SHOW	COM	#,Y	INVERT PIXELS THAT HAVE JUST BEEN DONE	
#3310	PSHS	CC	SAVE IT	#4310	LEAY	-#20,Y	POINT TO NEXT LINE UP	#4320	CMPLY	#BA	REACHED TOP OF PAGE?		
#3320	STA	STORE,PCR	SAVE SHIFTED BYTE	#4330	BPL	SHOW	IF NOT, GO DO MORE	#4340	TFR	U,Y	RESTORE Y		
#3330	CLRA		NOTHING IN ACC A IF PIXEL BLANK	#4350	RTS			#4360	EXTAB	FCB	#	TABLE TO EXPAND TWO BITS	
#3340	PULS	CC	GET FIRST PIXEL IN CARRY BIT	#4370	FCB	#07	INTO SIX	#4380	FCB	#38			
#3350	BCC	NONE	IF BLANK GO GET NEXT PIXEL	#4390	FCB	#3F		#4400	TITLE	FCC	/	SUPER/	
#3360	LDA	#0E0	EXPAND PIXEL INTO TOP THREE BITS	#4410	FCB	#00		#4420	FCB	/	GRAPHICS SCREEN/		
#3370	NONE	ROLB	GET SECOND PIXEL										
#3380	BCC	NONE2	GO PRINT IT IF BLANK										
#3390	ADDA	#91C	EXPAND INTO THREE BITS IF PIXEL SET										
#3400	NONE2	JSR	[CHROUT]	OUTPUT GRAPHICS TO PRINTER									
#3410	JSR	[CHROUT]											
#3420	JSR	[CHROUT]	FOUR TIMES										
#3430	JSR	[CHROUT]											
#3440	TST	EXPAND,PCR	DO WE WANT 5X EXPANSION?										
#3450	BEQ	SKP	NO, SKIP IT										
#3460	JSR	[CHROUT]	FIFTH TIME										
#3470	SKP	INC	BCOUNT,PCR	KEEP COUNT OF PIXELS IN CURRENT BYTE									
#3480	BNE	AGAIN	DO MORE IF NOT FINISHED										
#3490	COM	-#20,X	SHOW USER WHERE WE ARE										
#3500	COM	,X+	BY INVERTING THE PICTURE										
#3510	INC	LCOUNT,PCR	KEEP COUNT OF BYTES DONE										
#3520	BNE	NXTB	GO DO MORE IF NOT FINISHED										
#3530	NXTLT	CMPL	MID,PCR	DONE A PAGE?									
#3540	LBNE	NXTL	GO DO MORE IF NOT FINISHED										
#3550	CMPL	ENDG,PCR	DO SECOND MODE 4 PAGE?										
#3560	BEQ	DONE	GO IF FINISHED										
#3570	PSHS	X	SAVE POINTER TO CURRENT BYTE										
#3580	LDA	MID,PCR	GET MSB OF NEXT PAGE										
#3590	LDB	ENDG,PCR	SET MID TO END, SO WE DON'T DO THIS AGAIN										
#3600	STX	MID,PCR											
#3610	LBSR	DISP2	DISPLAY SECOND PAGE										
#3620	PULS	X	RESTORE POINTER										
#3630	LDBA	NXTL	GO DO NEXT PAGE										
#3640	DONE	LEAX	RESDAT,PCR	POINT AT RESET PRINTER CODES									
#3650	LBSR	PRTD	SEND TO PRINTER										
#3660	TST	INVERT,PCR	ARE GRAPHICS PAGES INVERTED?										
#3670	LBNE	EXIT	RETURN IF NOT										
#3680	LBSR	INV	LEAVE PAGES AS THEY STARTED										
#3690	LDBA	EXIT	FINISHED										

```

04430 FCB 000
04440 FCC / PRINTER DUMP/
04450 FCB 000
04460 FCC / BY/
04470 FCB 000
04480 FCC / JEFF LARSEN/
04490 FDB 00000
04500 FCB 0
04510 ERRMSG FCC / PRINTER OFF/
04520 FDB 00000
04530 FCC / RETURN TO BASIC?/
04540 FDB 00000
04550 FCC * YES OR NO (Y/N)*
04560 FDB 00000
04570 FCB 0
04580 QUEST1 FCC / SIZE OF DUMP/
04590 FDB 00000
04600 FCC / 1. SMALL/
04610 FCB 000
04620 FCC / 2. MEDIUM/
04630 FCB 000
04640 FCC / 3. MEDIUM
(EXPANDED)/
04650 FCB 000
04660 FCC / 4. LARGE/
04670 FCB 000
04680 FCC /
04690 FCB 0
04700 QUEST2 FCC / ONE OR TWO PAGES
(1 OR 2) /
04710 FCB 0
04720 QUEST3 FCC * INVERTED SCREEN?
(Y/N) *
04730 FCB 0
04740 WARN FDB 00000
04750 FCC / WARNING - YOUR SCREEN
DUMP CONTAINS LARGE
AMOUNTS OF BLACK,
WHICH MAY DAMAGE YOUR
PRINT HEAD./
04760 FDB 00000
04770 FCC * CONTINUE (Y/N)?*
04780 FCB 0
04790 RESDAT FDB 01032
04800 FDB 000FF
04810 OPSET FDB 01033
04820 FDB 013FF
04830 OPD FDB 00010
04840 FDB 04000
04850 FDB 001FF
04860 PRT20 FDB 00010
04870 FDB 0330C
04880 FCB 0FF
04890 TW001 FDB 00010
04900 FDB 04C00
04910 FDB 003FF
04920 TW002 FDB 00010
04930 FDB 04C00
04940 FDB 004FF
04950 PRT30 FDB 01033
04960 FDB 010FF
04970 PGRAP3 FDB 02020
04980 FDB 02020
04990 FDB 01040
05000 FDB 04002
05010 FCB 0FF
05020 CMDST RMB 1 VARIABLE STORAGE
05030 ENMG RMB 2
05040 INVERT RMB 1
05050 STORE RMB 0
05060 BCOUNT RMB 1
05070 FLAG RMB 1
05080 MID RMB 2
05090 EXPAND RMB 1
05100 LCOUNT RMB 1
05110 ZZZZ EQU 0
05120 END START

```

HINT.....

CoCo 3 POKEs

Here are some CoCo 3 POKEs some people might find interesting ... Look at the table below and find the type of screen you like to look at, note the number (denoted as 'xx') and then type in ...

POKE359,57:POKE65314,xx

Screen type:

- 0: Normal screen
- 8: Black on orange
- 16: Black on green, lowercase
- 24: Black on orange, lowercase
- 32: Green on black
- 40: Orange on black
- 48: Green on black, lowercase
- 56: Orange on black, lowercase
- 64: Black on green, no border
- 72: Black on orange, no border
- 80: Black on green, no border, lowercase
- 88: Black on orange, no border, lowercase
- 96++: same as above, with exception of 128++, setting screen in different modes of screens.

TURTLE GRAPHICS

continued from page 15

```

128 IF(PEEK(65314) AND 1)=1THENP
RINT"PRINTER IS OFF-LINE!!":GOTO
21
129 FORT=1TOJ-1
130 PRINT#-2,T;"");A$(T)
131 NEXTT
132 GOTO21
133 '
134 IFERNO=20THENPRINT"I/O ERROR
":CLOSE:GOTO38
135 IFERNO=8THENPRINT"TOO MUCH D
ATA. TRY INCREASING THE DIMENS
ION NUMBERS IN LINE 16 ":J=999:G
OTO21
136 IFERNO=14 OR ERNO=13 OR ERNO
=15THENPRINT"STRING TOO LONG":GO
TO140
137 IFERNO=5THENPRINT"NUMBER TOO

```

```

BIG":GOTO140
138 IFERNO=21 THEN PRINT"BAD FIL
E DATA":GOTO38
139 IFCH<0 OR CH>320 OR CV<0 OR
CV>192 THENPRINT"TURTLE WENT OFF
THE SCREEN"
140 PRINT"ERROR IN"W
141 GOTO38
142 IFJ=1THEN21ELSEG$=A$(J)
143 MR=INSTR(2,G$,"-")
144 IFMR=0THEN21
145 ST=VAL(MID$(G$,2,LEN(G$)-MR)
)
146 EN=VAL(MID$(G$,MR+1,LEN(G$)-
MR))
147 IFEN>J THENPRINT"ILLEGAL NUM
BER":GOTO21
148 IFAS("<M")THEN154
149 FORT=ST TOEN
150 A$(J)=A$(T)
151 J=J+1
152 NEXTT
153 GOTO21
154 FORT=ST TOEN+(J-EN)+1
155 A$(T)=A$(T+(EN-ST)+1)
156 NEXTT
157 J=J-(EN-ST)-1
158 IFJ<1THENJ=1
159 GOTO21
160 '
161 P1=INSTR(1,B$,"")

```

```

162 CH=VAL(LEFT$(B$,P1-1))
163 CV=VAL(RIGHT$(B$,LEN(B$)-P1)
)
164 GOTO69
165 J=J+1:IF(PEEK(65314)AND1)=1T
HENPRINT"PRINTER IS OFF-LINE!!"
:GOTO21
166 PRINT#-2,CHR$(18)CHR$(27)CHR
$(14);
167 FORT=0TO195STEP7
168 FORX=0TO320
169 G2=128
170 FORC=0TO6
171 IFY2+C8<192ANDHPOINT(X2,Y2+C
8)=0THENG2=G2+2^C8
172 NEXTC8
173 PRINT#-2,CHR$(G2);
174 NEXTX2
175 PRINT#-2
176 NEXTY2
177 PRINT#-2,CHR$(30)
178 GOTO21
179 B2=VAL(B$):IFB2>=J THENPRINT
"ILLEGAL NUMBER":GOTO21
180 PRINTB2";A$(B2)
181 PRINTB2";:LINEINPUTA$(B2)
182 GOTO21
183 ' A PRODUCT OF:
184 ' MERL SOFTWARE
185 ' 12/12/86

```

BASE GENERATOR

by Justin Lipton
16K CB

BASE GENERATOR allows you to enter a number in bases 2 to 10 or just base 16 and it will convert them to decimal. Binary, Octagonal, & hexadecimal are all accepted. I hope many people find it a handy utility.

The Listing:

```
0 GOTO5
1 '***** BASE GENERATOR *****
2 '***** JUSTIN LIPTON *****
3 SAVE"107:3":END
5 DIM S(10):REM ACCEPTS BINARY(B
ASE 2), OCTECIMAL(BASE 8),DECI
MAL (BASE 10),HEXADECIMAL(BASE 1
6) AND BASES 3,4,5,6,7,9
10 F=0
20 CLS
30 PRINT " BASE CALCULATOR
"
40 PRINT"ENTER WHAT BASE YOU WAN
T CONVERTED INTO DECIMAL
BASES ACCEPTED (2-10) O
R 16(HEX)";
50 PRINT"FOR EXAMPLE; TO ENTER A
BINARY NUMBER HIT 2 enter THEN
INPUT YOUR NUMBER AND IT WIL
L BE CONVERTED TO DECIMAL."
60 B=0:S(1)=0:F=0:N=0
70 PRINT @ 288, ""::INPUT "WHAT
BASE ";B
80 IF B=16 THEN 100
90 IF B<2 OR B>10 THEN 70
100 PRINT"INPUT NUMBER IN BASE"B
::INPUT A$
110 L=LEN(A$)
120 FOR E=1 TO L
130 N$=MID$(A$,E,1)
140 IF B=16 AND N$="A" THEN N=10
:GOTO 220
150 IF B=16 AND N$="B" THEN N=11
:GOTO 220
160 IF B=16 AND N$="C" THEN N=12
:GOTO 220
170 IF B=16 AND N$="D" THEN N=13
:GOTO 220
180 IF B=16 AND N$="E" THEN N=14
:GOTO 220
190 IF B=16 AND N$="F" THEN N=15
:GOTO 220
200 N=VAL(N$)
210 IF N>B-1 THEN PRINT"WRONG BA
SE ";:FOR T=0 TO 400:NEXT T:RUN
220 S(E)=N*B^(L-E)
230 NEXT E
240 FOR E=1 TO L
250 F=F+S(E)
260 NEXT E
270 PRINT "ANSWER IN DECIMAL ="F
280 INPUT "GO AGAIN (Y/N)";Y$
290 IF Y$<>"Y" THEN CLS:END: ELS
E RUN
```

MEMORY LOOK

by Gregory Becroft

16K ECB
UTILITY

MEMORY LOOK ALLOWS you to look at any part of memory on the graphic screen in any PMODE of the Text screen.

I find it a great plus in understanding the memory map of the CoCo by seeing the machine in actual operation.

It uses both joysticks. The right for scanning through the memory pages and the left for selecting various page and screen modes. The right joystick button switches between the scanning mode and the select mode.

It uses the screen address switches at locations 65478 to 65491 via the right joystick for the scanning and a subroutine for changing screen modes with the left joystick. It also displays which page location you are at and if you don't move the left joystick you can see the modes you where in last.

On examination of memory, points of interest are the scratch pad at the start of memory in which you can see various locations changing, such as the keyboard rollover table (by typing) and the joysticks (by moving the left joystick). If you have a 64K CoCo then you can see the Basic ROM's when in the 64K RAM mode. You can see the CLEAR statement in operation by looking just before where the Extended Basic ROM starts, breaking out of the program, resetting CLEAR (eg CLEAR 400) then re-RUNing the program. Other programs can be seen by loading them in before loading MEMLOOK. Other areas to search for are the text screen, disk drive scratch pad and looking at graphics screens that you have previously drawn (use PCLEAR 8 as well).

The more I play with this program the more I discover about the CoCo memory, so have fun experimenting.

The Listing:

```
0 '***** MEM LOOK *****
1 '***** GREGORY J BECROFT ***
2 GOTO10
3 SAVE"47:3":END
10 A=0:C=1:D=1
20 PMODE0,1
30 SCREEN 1,1
40 CLS
50 P=PEEK(65280):IFP=1260RP=254T
HEN GOSUB 320
60 P1=INT(LOYSTK(0)/4)
70 P2=INT(JOYSTK(1)/4)
80 PS=P1*8+P2
90 PN=PS
100 IFPS<64 THEN POKE 65490,0
110 IFPS>63 THEN POKE 64591,0
120 IFPS>63 THEN PS=PS-64
130 IFPS<32 THEN POKE65488,0
140 IFPS>31 THEN POKE 65489,0
150 IFPS>31 THEN PS=PS-16
160 IFPS<16 THEN POKE65486,0
170 IFPS>15 THEN POKE 65487,0
180 IFPS>15 THENPS=PS-16
190 IFPS<8THEN POKE65484,0
200 IFPS>7THEN POKE 65485,0
210 IFPS>7 THEN PS=PS-8
220 IFPS<4THEN POKE65482,0
230 IFPS>3THEN POKE 65483,0
240 IFPS>3THEN PS=PS-4
250 IFPS<2THEN POKE65480,0
260 IFPS>1THEN POKE65481,0
270 IFPS>1THEN PS=PS-2
280 IFPS<1THEN POKE65478,0
290 IFPS>0THEN POKE 65479,0
300 IFPS>0THEN PS=PS-1
310 GOTO40
320 CLS
330 PRINT"YOU ARE LOOKING AT PAG
E"
340 PRINTPN;"DECIMAL"
350 PRINT " ";HEX$(PN);"HEXIDECIM
AL"
360 PRINT@231,"PMODE ,1 SCREEN
"
370 IFA=0THEN POKE1260,48:IFA=1T
HEN POKE1260,49:IFA=2THEN POKE12
60,50:IFA=3THEN POKE1260,51:IFA=
4THEN POKE1260,52
380 IPC=0THEN POKE1271,48ELSEPOK
E1271,49
390 IPD=0THEN POKE1273,48ELSEPOK
E1273,49
400 Z=JOYSTK(0):Y=JOYSTK(2)
410 X=JOYSTK(3)
420 IFY<32 AND X<12 THEN A=0:POK
E1260,48
430 IFY<32ANDX>11 ANDX<24THEN A=
2:POKE1260,50
440 IFY<32ANDX>23 ANDX<38THEN A=3
:POKE 1260,21
```


Draw your own conclusions
with CoCo 3 graphics.....



COCO 3 DRAWING MACHINE

CoCo 3 + Joystick
GRAPHICS UTILITY

by Andrew McLintock

AWHILE AGO Dad got himself one of these flashy new colour computer three's. At first I didn't want too much to do with it, probably because I didn't understand half the new commands. After trying a few different ideas I soon found out that the graphics are very impressive. That's what Drawing Machine is all about, a way of using the graphics in the easiest way possible. Nearly all the commands can be controlled by the joystick and this was the best way I found to make Drawing Machine easy to use.

There is a complete set of instructions with the programme but if they look to be a bit long to put in and you don't want them then DEL 1150-1360.

You also need to change these lines:- 1140 RETURN. Get rid of the last HPRINT command in 1130. It is important that line 1370 stays the same line number.

Line 10 is needed if you have BDOS. You can save your drawings on tape or disk but you will need a Coco 3 and a joystick.

The Listing:

```
0 GOTO 10
1 '**DRAWING MACHINE**
2 '**BY ANDREW MCLINTOCK**
3 SAVE"148:3":END
10 POKE &H13E,0:POKE&H143,0 'FOR
BDOS.
20 Y=1:HBUFF 4,3700:GOSUB 1110
30 PCLEAR 6:PON 2:HBUFF 1,66:HBU
FF 2,66:HBUFF 5,3700:C1=1:C2=1:U
=1
40 OS9 0,27:OS9 1,12:OS9 3,20:OS
9 4,0:OS9 5,36:OS9 6,32:OS9 7,18
:OS9 8,55:OS9 9,63:OS9 10,44:OS9
11,40:OS9 12,45:OS9 13,56:OS9 1
4,2:OS9 15,43
50 DUMP 2,0:HLINE(105,100)-(100,
100),PSET:HLINE-(100,105),PSET:H
DRAW"BM104,102;F5G2H5G2E3C0E1C2E
3":MON(101,101),2,2:DUMP 1,0
60 HGET(100,100)-(110,110),1:PON
2
70 GOSUB 300
80 A=128:B=96:A1=A:B1=B
```

```
90 HGET(A,B)-(A+10,B+10),2
100 HPUT(A,B)-(A+10,B+10),1
110 A1=A:B1=B:A=JOYSTK(0)*5:B=JO
YSTK(1)*3
120 IF K=6 AND PD=1 THEN GOSUB93
0:HLINE-(A,B),PSET:GOSUB940
130 IF A>4 AND A<46 AND B<190 AN
D B>130 AND BUTTON(0)=1 THEN GOS
UB930:C=HPOINT(A,B):GOSUB940:IF
K=4 THEN 600 ELSE IF K=7 THEN GO
SUB 700
140 IF BUTTON(0)=1 AND A<52 THEN
GOSUB 910:GOTO 210
150 IF BUTTON(0)=1 AND B<182 AND
A>52 THEN GOSUB 920:ON K GOSUB
440,470,500,580,640,670,700,800,
820:GOTO 180
160 IF K=4 AND A>52 AND B>182 AN
D BUTTON(0)=1 THEN 580
170 IF K=7 AND A>241 AND B>181 A
ND BUTTON(0)=1 THEN 730
180 IF A=A1 AND B=B1 THEN 110
190 HPUT(A1,B1)-(A1+10,B1+10),2
200 T=0:GOTO 90
210 IF A>44 THEN 580
220 IF B<4 OR B>124 THEN 180
230 IF A>4 AND A<24 THEN 270
240 B2=INT((B-4)/20):B2=B2+1
250 ON B2 GOSUB 630,660,690,750,
790,810
260 GOTO 180
270 B2=INT((B-4)/20):B2=B2+1
280 ON B2 GOSUB 430,460,490,520,
560,570
290 GOTO 180
300 FOR X=4 TO 40 STEP 20:FOR Y=
4 TO 119 STEP 20
310 HLINE(X,Y)-(X+20,Y+21),PSET,
B
320 NEXTY:NEXTX
330 FOR X=4 TO 39 STEP 10:FOR Y=
130 TO 189 STEP 15
340 HLINE(X,Y)-(X+10,Y+15),PSET,
B
350 MON(X+1,Y+1),C,1:C=C+1
360 NEXTY:NEXTX
370 C=1
380 HLINE(52,0)-(52,191),PSET:MO
N(2,2),1,1:FORMAT(14,14),7:HLINE
(28,8)-(40,20),PSET,B:HLINE(20,2
8)-(8,40),PSET:AS="U4E9F3G7H3F3G
2L4"390 HDRAW"BM28,41;" +AS:MON(3
2,35),1,1:AS="E5U4G5D4H6U4E5F6G5
H6":HDRAW"BM15,61;" +AS:AS="R8U8H
2U2L4D2G2R8L8D8":HDRAW"BM33,61;"
+AS:HLINE(27,51)-(35,51),PSET:HL
INE-(28,48),PSET:HLINE(28,54)-(3
5,51),PSET
400 HDRAW"BM8,81;R8E2U2H2L6H2U2E
2R8":HDRAW"BM28,81;U12D12R12":HD
RAW"BM 18,101;L8H2U8E2R8":HDRAW"
BM27,101;U10E1R3F1D5L5R5D5":HDRA
```

```
W"BM35,101;U11R4F1D3G1L4R4F1D4G1
L4"
410 HDRAW"BM9,121;U12R7F1D5G1L7"
:HDRAW"BM29,121;U10R4U3R2D3L3R7D
3L10R10D7L10":MON(34,109),1,1:HL
INE(50,192)-(320,182),PSET,B:HLI
NE(50,181)-(320,181),PSET
420 RETURN
430 K=1:L=0:HPRINT(18,23),"-CIRC
LE-":RETURN
440 IF L=1 THEN 450 ELSE GOSUB93
0:HSET(A,B,U):GOSUB940:L1=A:L2=B
:L=1:RETURN
450 R=SQR(((A-L1)^2)+((B-L2)^2))
:GOSUB930:FORMAT(L1,L2),R,U:GOSU
B940:L=0:HRESET(L1,L2):K=0:GOSUB
910:RETURN
460 K=2:L=0:HPRINT(19,23),"-LINE
-":RETURN
470 IF L=1 THEN 480 ELSE GOSUB93
0:HSET(A,B,U):GOSUB940:L1=A:L2=B
:L=1:RETURN
480 GOSUB930:HLINE(A,B)-(L1,L2),
PSET:GOSUB940:L=0:RETURN
490 K=3:L=0:HPRINT(18,23),"-ERAS
ER-":RETURN
500 IF L=1 THEN 510 ELSE GOSUB93
0:HSET(A,B,U):GOSUB940:L1=A:L2=B
:L=1:RETURN
510 GOSUB930:DUMP 0,1:HLINE(A,B)
-(L1,L2),PSET,BF:GOSUB940:DUMP U
,0:RETURN
520 HPRINT(8,23),"SAVE PICTURE (
Y/N)":GOSUB1370:IF AS="Y"THEN 53
0 ELSE GOSUB910:RETURN
530 GOSUB910:HPRINT(8,23),"SAVE
TO (D)ISK OR (T)APE
540 GOSUB1370:IF AS="D" OR AS="T
" THEN D$=AS:GOTO 550 ELSE 540
550 GOSUB910:HPRINT(8,23),"FILEN
AME-":A2=17:GOSUB890:GOSUB930:GO
SUB950:GOSUB1000:GOSUB940:GOSUB9
10:RETURN
560 HPRINT(8,32),"-CLEAR- ARE YO
U SURE(Y/N)":GOSUB1370:IF AS="Y"
THEN DUMP 0,5:HLINE(53,0)-(320,
180),PSET,BF:DUMP 1,0:GOSUB910:R
ETURN ELSE RETURN
570 K=4:HPRINT(11,23),"COLOUR":H
PRINT(28,23),"PALETTE":HLINE(182
,192)-(182,182),PSET:RETURN
580 IF A<182 OR L=1 THEN 600 ELS
E GOSUB 910:HPRINT(10,32),"SLOT-
":A2=15:GOSUB 890:IF VAL(B$)<0 O
R VAL(B$)>15 THEN A=190:B$="":GO
TO 580
590 P1=VAL(B$):B$="":HPRINT(23,2
3),"COLOUR-":A2=30:GOSUB 890:IF
VAL(B$)<0 OR VAL(B$)>63 THEN B$=
"":GOTO 580 ELSE OS9 P1,VAL(B$):
B$="":GOSUB 910:GOTO 180
600 GOSUB910:IF L=1 THEN 610 ELS
```

```

E HPRINT(18,23),"-COLOUR-":L=1:GOTO 180
610 HPRINT(16,23),"CORRECT (Y/N)":GOSUB1370:IF A$="N" GOSUB910:L=0:GOTO 180
620 U=C:L=0:GOSUB 910:DUMP U,0:GOTO 180
630 K=5:L=0:HPRINT(20,23),"-BOX-":RETURN
640 IF L=1 THEN 650 ELSE GOSUB930:HSET(A,B,U):GOSUB940:L1=A:L2=B:L=1:RETURN
650 GOSUB930:HLINE(A,B)-(L1,L2),PSET,B:GOSUB940:L=0:K=0:GOSUB910:RETURN
660 K=6:L=0:HPRINT(18,23),"-PENCIL-":RETURN
670 IF L=1 THEN 680 ELSE HLINE(A,B)-(A,B),PSET:L=1:PD=1:RETURN
680 PD=0:K=0:L=0:GOSUB910:RETURN
690 K=7:L=0:HPRINT(8,23),"CHANGE PAINT COLOURS- YES NO":HLINE(278,182)-(278,191),PSET:HLINE(241,182)-(241,191),PSET:RETURN
700 IF L=3 THEN 720 ELSE IF L=2 THEN 710 ELSE C1=C:L=2:GOSUB910:HPRINT(14,23),"-BOUNDRY COLOUR-":RETURN
710 C2=C:L=3:GOSUB910:HPRINT(18,23),"-PAINT-":RETURN
720 GOSUB930:MON(A,B),C1,C2:GOSUB940:L=0:K=0:GOSUB910:RETURN
730 IF A<278 THEN 740 ELSE GOSUB910:HPRINT(18,23),"-PAINT-":L=3:GOTO 180
740 GOSUB910:HPRINT(15,23),"-PAINT COLOUR-":L=1:GOTO 180
750 HPRINT(8,23),"LOAD PICTURE (Y/N)":GOSUB1370:IF A$="Y" THEN 760 ELSE GOSUB910:RETURN
760 GOSUB910:HPRINT(8,23),"LOAD FROM (D)ISK OR (T)APE":770 GOSUB1370:IF A$="D" OR A$="T" THEN D$=A$:GOTO 780 ELSE 770
780 GOSUB910:HPRINT(8,23),"FILE NAME-":A2=17:GOSUB890:GOSUB950:GOSUB1040:GOSUB910:RETURN
790 K=8:HPRINT(7,32),"*":A2=8:GOSUB890:HPRINT(A2,23),"*":RETURN
800 GOSUB930:HPRINT(A/8,B/8),B$:GOSUB940:B$="":K=0:GOSUB910:RETURN
810 K=9:HPRINT(17,23),"CUT & PASTE":L=0:RETURN
820 IF L>0 THEN 830 ELSE GOSUB930:HSET(A,B,U):GOSUB940:L=1:L1=A:L2=B:RETURN
830 IF L=2 THEN 860 ELSE G1=ABS(A-L1):G2=ABS(L2-B):G=INT(G1/2)+1:IF G2>G>3700 THEN RETURN ELSE:GOSUB930:HGET(A,B)-(L1,L2),4:HLINE(A,B)-(L1,L2),PSET,B:GOSUB910:HPRINT(16,23),"CORRECT (Y/N)"
840 GOSUB1370:IF A$="N" THEN HPUT(A,B)-(L1,L2),4:GOSUB940:GOSUB910:L=0:HRESET(L1,L2):RETURN
850 L=2:HPUT(A,B)-(L1,L2),4:HRESET(L1,L2):GOSUB940:GOSUB910:HPRINT(15,23),"-POSITION OBJECT-":RETURN
860 GOSUB930:HGET(A,B)-(A+G1,B+G2),5:HPUT(A,B)-(A+G1,B+G2),4:GOSUB910:HPRINT(16,23),"CORRECT (Y/N)"
870 GOSUB1370:IF A$="N" THEN HPUT(A,B)-(A+G1,B+G2),5:GOSUB940:GOSUB910:L=0:RETURN

```

```

880 L=0:GOSUB940:GOSUB 910:RETURN
890 A$=INKEY$:IF A$="" THEN 890 ELSE IF A$=CHR$(13) THEN RETURN
900 B$=B$+A$:HPRINT(A2,32),A$:A2=A2+1:IF A2>39 THEN RETURN ELSE 890
910 DUMP 0,1:HLINE(53,183)-(320,192),PSET,BF:DUMP U,0:RETURN
920 IF BUTTON(0)=1 THEN 920 ELSE RETURN
930 HPUT(A1,B1)-(A1+10,B1+10),2:RETURN
940 HGET(A1,B1)-(A1+10,B1+10),2:HPUT(A1,B1)-(A1+10,B1+10),1:RETURN
950 RESTORE:D=0:FOR X=&HE00 TO &HE44:READ E$:E=VAL("&H"+E$):D=D+E:POKE X,E:NEXTX:IF D<>&H183C THEN PRINT"ERROR IN DATA STATEMENTS":STOP
960 POKE &HE6E1,&H7E:POKE &HE6E2,&H0E:POKE &H0E02,&H20:IF D$="T" THEN GOSUB 910:HPRINT(8,23),"SET UP TAPE RECORDER."
970 IF D$="D" THEN GOSUB910:HPRINT(8,23),"PRESS <ENTER>"
980 GOSUB 1370
990 RETURN
1000 POKE &H0E04,0:FOR X=1 TO 8:N1$=B$+MID$(STR$(X),2):AUTO:IF D$="D" THEN SAVEMN1$,&H1000,&H1FFF,0:GOTO 1020
1010 CSAVE N1$,&H1000,&H1FFF,&H1000
1020 NEXTX
1030 B$="":POKE &HE6E1,&H8E:POKE &HE6E2,&H20:RETURN
1040 POKE &H0E04,1:FOR X=1 TO 8:N1$=B$+MID$(STR$(X),2):IF D$="D" THEN LOADM N1$ ELSE CLOADM N1$
1050 AUTO:NEXTX:GOTO 1030
1060 DATA 20,0D,20,0,0,10,0,20,0,0,0,0,0,0
1070 DATA 30,8C,F0,EF,07,10,AF,09,ED,0B,1F,13
1080 DATA 10,AE,40,AE,43,6D,42,26,17,A6,A0,A7,80,AC,45,26,F8,10,AF,40
1090 DATA 1F,31,EE,07,10,AE,09,EC,0B,7E,E6,EB
1100 DATA A6,80,A7,A0,AC,45,26,F8,20,E7
1110 RAM 40:OS9 11,36:OS9 12,63:PON 2:OS9 0,63:OS9 1,0:P$="L20U3R20D3U18L20D15U15E5R10F5H5U3L2D3L2U3L2D3L2U3L2D3U3R10":HDRAW"BM30,30"+P$:HLINE(10,23)-(30,15),PSET,B:HGET(10,3)-(30,30),4:FOR X=10 TO 320 STEP 40:HPUT(X,3)-(X+20,30),4
1120 HPUT(X,190)-(X+20,163),4:OS9 0,63:DUMP Y,0:HLINE(X+1,22)-(X+19,16),PSET,BF:HLINE(X+1,182)-(X+19,176),PSET,BF:Y=Y+1:NEXTX:HPRINT(16,5),"COCO 3"
1130 HPRINT(11,9),"DRAWING MACHINE":HPRINT(18,17),"By":HPRINT(11,19),"Andrew McIntock":HPRINT(13,13),"Instructions"
1140 GOSUB 1370:IF A$="Y" THEN 1160 ELSE RETURN
1150 ATTR 5,4
1160 RAM 80:Q$="-INSTRUCTIONS-":LOCATE 33,0:PRINTQ$:LOCATE 2,2:PRINT"Drawing Machine is a Hi-Res graphic's programme and require

```

a single joystick."ATTR 3,4:LOCATE 0,4:PRINT"Display-":ATTR 5,4:LOCATE 8,4

1170 PRINT"The screen is divided into four areas, these are:-The programme commands, Colour grids, Drawing page and the Operation display. The commands are shown in to rows of six squares each with its own diagram. The colour grid is below the"

1180 LOCATE 0,7:PRINT" commands and are shown as sixteen different coloured squares. The drawing page is the large square in the middle of the screen. The operations display is the rectangle at the bottom of the screen and it gives all messages and"

1190 LOCATE 67,9:PRINT"information. The cursor is the small arrow and is controlled by the right joystick. To select a command position the arrow over the symbol and press the fire button."ATTR 7,4:LOCATE 0,12:PRINT"PROGRAMME COMMANDS":ATTR 3,4

1200 LOCATE 0,13:PRINT"Circle-":ATTR 5,4:LOCATE 7,13:PRINT"Draws a circle: Position arrow and press button, a small dot will show the centre of the circle. Move the arrow and push button and a circle will be drawn with a radius distance of the arrow and"

1210 LOCATE 42,15:PRINT"the centre. The small dot will disappear."ATTR 3,4:LOCATE 0,17:PRINT"Box-":ATTR 5,4:LOCATE 5,17:PRINT"Draws a box: Position arrow and press button, a small dot will mark one corner. Move arrow to other corner and press button."

1220 ATTR 3,4:LOCATE 0,19:PRINT"Line-":ATTR 5,4:LOCATE 5,19:PRINT"Draws a line: Position arrow and press button and one end of the line will be marked. Position arrow at the other end and press button."ATTR 3,4:LOCATE 0,21:PRINT"Pencil-":ATTR 5,4

1230 LOCATE 8,21:PRINT"Draws a continuous line: Press button to lower pencil. Draw with joystick and press button to lift it."GOSUB 1360:CLS 5:LOCATE 33,0:ATTR 5,4:PRINTQ\$:ATTR 3,4:LOCATE 0,1:PRINT"Eraser-":ATTR 5,4:LOCATE 7,1

1240 PRINT"Clears a certain area: Works the same as box command but uses background colour."ATTR 3,4:LOCATE 0,3:PRINT"Paint-":ATTR 5,4:LOCATE 7,3:PRINT"Paints an enclosed area: When you select this command you have the option to change the paint"

1250 LOCATE 21,4:PRINT"and boundary colours or to use the previous colours set. To select new options place the arrow over the box marked <Yes> and press button. Move the arrow to the colour grid and place it over the colour you want to paint with and"

1260 LOCATE 9,7:PRINT"press button. Do the same for the boundry

colour. Move the cursor to the Drawing page and press button. The screen will be coloured the paint colour until it hits the boundary colour. If you wish to use the same colours selected the"

1270 LOCATE 1,10:PRINT" box marked (No) and position the arrow on the Drawing page and press button.":ATTR 3,4:LOCATE 0,11:PRINT"Save-":ATTR 5,4:LOCATE 6,11:PRINT"Allows you to save a picture to tape or disk.":ATTR 3,4:LOCATE 0,12:PRINT"Load-"

1280 LOCATE 6,12:ATTR 5,4:PRINT" Allows you to load a picture from tape or disk.":ATTR 3,4:LOCATE 0,13:PRINT"Clear-":ATTR 5,4:LOCATE 6,13:PRINT"CLEAR"Clears the drawing page.":ATTR 3,4:LOCATE 0,14:PRINT"Letters-":ATTR 5,4:LOCATE 8,14

1290 PRINT" Draws letters on the drawing page: Type in message from keyboard and it will appear on the operation display. Press (Enter) when finished. Position the arrow on the screen and press button. The message will appear on the drawing page."

1300 ATTR 3,4:LOCATE 0,18:PRINT" Palette-":ATTR 5,4:LOCATE 9,18:PRINT"Controls drawing colours and Palette changes: To change drawing colour place arrow over box marked (Colour) and press button. Then move the arrow to the colour grid and select a colour.

1310 LOCATE 35,20:PRINT"All messages will appear in the drawing colour.":GOSUB 1360:CLS 5:LOCATE 33,0:ATTR 5,4:PRINTQ\$:LOCATE 0,1

1320 ATTR 3,4:PRINT"Cut and Paste-":LOCATE 14,1:ATTR 5,4:PRINT"Allow you to get an area of the screen and put that image elsewhere: Position arrow and press button, this marks the one corner of the area. Move arrow and press button and a box will"

1330 LOCATE 37,3:PRINT" mark the area to be moved. Reposition arrow and press button and the image is put back.":ATTR 3,4:LOCATE 0,5:PRINT"Colour Grid:-"

1340 LOCATE 0,6:ATTR 5,4:PRINT"Whenever ever selecting a colour on the colour grid make sure that the point of the arrow is well on the square. The first box holds the background colour, the second is the foreground colour (at the start of the"

1350 LOCATE 48,8:PRINT"programme). The third holds the arrow colour.":LOCATE 1,10:PRINT"The programme has several other bits and pieces but these are not major and you will find them as you go along. Have Fun !!!":GOSUB 1360:ATTR 3,4:RETURN

1360 ATTR 3,4:LOCATE 25,23:PRINT"PRESS (ENTER) TO CONTINUE.":ATTR 5,4

1370 A\$=INKEY\$:IF A\$="" THEN 1370 ELSE RETURN

DISK FIX for COCO 3



DISK UTILITY

by Fred Bisseling

THE PROGRAM called DFIX/BAS has not been altered and is used to set up Double Sided Drives (if you have them) as well as increasing the drive speed. Hope this is of use for someone out there!

The Listing:

```
0 GOTO10
1 ***** DFIX *****
***** FRED BISSELING *****
3 SAVE"140A:3":END
10 CLEAR 200,28672
20 FOR I= 28672TO 29423
30 READ Q$:POKEI,VAL("&H"+Q$):NEXTI
40 INPUT"SAVE TO TAPE OR DISK":A$:IF A$="T" THEN CSAVE"DFIX",28672,29423,28672 ELSE SAVE"DFIX":28672,29423,28672
50 DATA 2,FF,81,53,27,1A,7E,C9,E0,4,20,1C,0,0,10,20,20,20,18,2,20,1C,0,0,4,20,20,20,18,48,52,53,9D,9F,96,E6,27,56,30,8C,F4,8D,54,8D,55,96,E6,8D,54,B6,E6,78,8D,4F,96,E6,4A,C6,5,3D,33,8C,CA,33,C5,E6,C0,8E,E6,78,A6,80,8D,3B,5A
60 DATA 26,F9,86,30,34,42,8E,20,0,A6,C4,5F,1F,2,F6,FF,A1,A6,E4,1A,50,B7,FF,A1,A6,80,F7,FF,A1,1C,EF,8D,19,31,3F,26,E9,35,42,4C,E6,C0,C1,20,27,D6,7E,A4,2D,7E,B4,4A,7E,C9,38,7E,CA,A,7E,CC,24,7E,CA,7,7E,C5,C4,7E,C3,52,81,53,27,3,7E
70 DATA CA,48,9D,9F,30,8C,80,8D,E0,8D,E7,8D,E8,1F,89,4F,BD,E6,96,8D,E0,B7,FF,9A,96,E6,33,8D,FF,55,4A,C6,5,3D,33,C5,E6,C0,8E,FF,B0,10,8E,E6,78,8D,C6,A7,80,A7,A0,5A,26,F7,86,30,34,42,8E,20,0,A6,C4,5F,1F,2,8D,B1,D6,70,26,B0,F6,FF,A1
80 DATA 34,4,E6,61,1A,50,F7,FF,A1,A7,80,35,4,F7,FF,A1,1C,EF,31,3F,26,E1,35,42,4C,E6,C0,C1,20,27,CE,7E,A4,2D,B6,1,34,81,D,27,23,9E,A6,34,10,BD,B1,56,8E,1,A9,9F,B,35,10,D,6,2B,F,BD,B3,ED,10,83,0,3,22,3,D7,EB,39,7E,A6,1F,9F
90 DATA A6,8E,9,54,16,FF,4C,8D,F,8E,0,0,30,1F,27,7,B6,FF,48,85,1,26,F5,39,34,5,C6,F,5A,26,FD,35,85,C4,1,3A,E6,84,39,D6,EB,C4,1,6F,85,39,C1,FB,C2,5,D3,CC,D7,D1,C9,38,CA,4,CC,24,CA,7,C5,C4,C3,52,D7,8C,D7,F1,D7,C0,D8,16,D7
```

100 DATA 7D,D6,E,D6,17,D7,C4,D8,1A,D7,E4,D8,51,D8,4,D7,BB,D8,9F,C1,E5,C1,EF,D2,DF,D6,EC,C8,8A,C9,56,CB,52,C9,59,C5,97,C3,34,D6,99,D6,FE,D6,CD,D7,23,D5,21,D6,8A,D5,2A,D6,D1,D7,27,D6,F1,D7,5E,D7,11,D6,C8,D7,AC,6,8E,80,0,BD,A7,D3,2

110 DATA 11,17,1,0,1,14,1,0,1,0,0,6,8E,0,0,BD,A7,D3,2,22,2E,1,0,1,14,1,0,1,0,0,6,BD,A7,D1,BD,A7,D1,2,22,2E,1,1,1,15,1,0,1,0,0,6,BD,A7,D1,BD,A7,D1,2,44,5C,1,2,1,16,1,0,1,0,0,6,BD,A7,D1,BD,A7,D1,120 DATA 2,44,5C,1,3,1,17,1,0,1,0,0,34,1,1A,50,CE,71,5A,FC,C0,4,10,83,D7,5F,27,B,CE,71,8A,10,83,D6,6C,27,2,35,81,AE,C1,EC,84,10,83,D9,8F,27,F4,EC,84,FD,70,96,CC,D9,8F,ED,84,AE,C1,EC,84,FD,70,7,CC,D9,0,ED,84,CC,DA,1,ED

130 DATA D1,AE,C1,86,7E,A7,80,CC,DA,31,ED,84,C6,6,8E,70,80,10,AE,C1,10,AF,81,30,1,5A,26,F5,10,8E,71,BA,B6,70,0,81,4,23,2,86,4,C6,13,3D,31,AB,E6,A0,27,B,AE,C1,A6,A0,A7,80,5A,26,F9,20,F1,C6,5,10,AE,C1,8E,12,BD,AF,A1,8E,DA,42,A,F

140 DATA A4,5A,26,F0,B6,70,1,27,1C,AE,C1,86,BD,A7,80,CC,DA,4B,ED,84,AE,C1,CC,12,BD,ED,81,CC,DA,51,ED,84,CC,41,42,ED,D1,CE,70,2,8E,D9,0,10,8E,1,58,A6,C0,A7,80,31,3F,26,F8,CC,12,12,FD,E6,C6,B7,F,F,D9,35,81,0,0,0,0,0,0,0,0,0,0,0,0

150 DATA 0,0

MACHINE LANGUAGE PROGRAMMING

LESSON ONE
32K CoCo with EDTASM+

by Malcolm Patrick

I WOULD LIKE TO take this opportunity to welcome you to the joys and hard work of learning ASSEMBLY LANGUAGE PROGRAMMING for the 6809 Microprocessor. You will find at times that this subject can be boring and tedious. But I am sure it will also be very rewarding and as you learn more about the 6809 it will become more enjoyable and helpful in the day to day use of your computer.

I would advise the purchase of the book called "TRS-80 Colour Computer Assembly Language Programming" by Radio Shack cat no. 62-2077 as we will be referring to it from time to time.

You will of course also need the EDTASM+ ROM Pak that converts assembly language to machine language. It also comes on disk and both are available from Tandy.

I will start from the very beginning and take you through step by step. The first thing we do is plug in the ROM Pak and turn on the computer.

!!WARNING!! don't plug in the cartridge whilst the computer is switched on.

Once the EDTASM program is up and running you will see an asterisk (a "*"). This means that the computer is now waiting for a command.

We are going to be typing in Listing One, found below.

Press "I" and ENTER. (The "I" stands for Input.) You will see that line 100 has come up and is waiting for us to enter Assembly Code.

Start with the Asterisk and enter the remaining sentence, ie type '* RAINBOW CHECK PLUS IN HEXADECIMAL'.

Remember that any line starting with an Asterisk is a comment line and the words following it need not be entered, as it will only be ignored by the computer during assembly.

Next we come to line 130. You

will note that there are FIVE columns to this line. Column one is the line number that comes up automatically as you press ENTER; column two is the Label; column three is known as the Opcode; column four the Operand and column five the remark.

Coming back to Column two:

The word "ORG" is the OPCODE column (column two) and to get there we must press the right arrow key. Now type 'ORG'; press the right arrow again and type '\$3FB0'. Press the right arrow again and type '\$7FB0 FOR 32K'. This is a comment column and need not be entered. Now press ENTER, this completing line 130.

At line 140 you will note the word "START" - this is a label and we will type it straight in. Press the right arrow, type 'LDA', press right arrow again, type '\$016A', press ENTER. That completes Line 140.

If you make any mistakes, don't worry about them for now, just press ENTER.

Continue doing this procedure until you reach Line 520. Now press BREAK.

To look at the listing to see if there are any mistakes, type 'P#' and ENTER. This will take you back to the very first line, in our case Line 100. By entering 'P' we can look at the next 15 lines in the listing. By pressing the down arrow we can see the next line in the listing, up arrow for the preceding line.

Type 'P#:*' for the complete listing.

Let's say we found an error in line 140 and we wish to correct it. Enter "E140" to EDIT 140 and press the space bar till you reach the error. The editing procedures are the same as found in BASIC, with the exception of having to type "E140" instead of "EDIT 140".

After you have checked the listing for mistakes you may want to save it to tape. Prepare your tape recorder to record and

type 'W <filename>' and press ENTER. The 'W' stands for WRITE IN ASCII and the <filename> can be any word you wish to call the program.

Once the computer has saved your program, it can be called back into the Editor Assembler anytime by typing 'L <filename>'. The 'L' stands for 'LOAD'.

For Disk users that wish to save or read the listing to or from tape and not disk, add the letter 'C' immediately following 'W' or 'L' and prior to the space, i.e. if you want to load something from tape, type 'LC <filename>' or if you want to save something onto tape type 'WC <filename>'.

Only after we have saved the listing in ASCII can we save the listing to tape as a BINARY or MACHINE code program, the reason being that once it is saved in Binary it can't be read back into the assembler.

Set your tape recorder to record again and type 'A <filename>' and press ENTER. After a short period of time you will see a lot of numbers scrolling down the screen with the cassette recording.

Next, turn off your computer and unplug the ROM Pak. Turn it on again and type 'CLOADM filename' After it is loaded type in 'EXEC'.

Listing One came from the August 1983 AUSTRALIAN RAINBOW and is used to check for typing mistakes. It displays the count at the top left of the screen. See the front pages of "Australian Rainbow Magazine" for details.

For those of you who are interested, I have included a questionnaire for you to fill out at the end of each lesson. You can then see how you went. If you don't know the answer to a particular question, re-read the above.

If you want to check your work, the answers are given below the questions.

00120 *			
00130	ORG	\$3FB0	\$7FB0 FOR 32K
00140 START	LDA	\$016A	
00150	STA	RETURN,PCR	
00160	LDA	#\$7E	
00170	STA	\$016A	
00180	LDX	\$016B	
00190 VAR	STX	RETURN+1,PCR	
00200	LEAX	CHECK,PCR	
00210	STX	\$016B	
00220	RTS	BACK	
00230 CHECK	CMPA	#\$0A	DOWN ARW KEY
00240	BNE	RETURN	NOT PRESSED
00250	PSHS	X,D	SAVE REGISTERS
00260	LDD	<\$1B	TOP OF RAM
00270	SUBD	<\$19	START OF PRGM
00280	LDX	#\$400	START OF SCREEN
00290	BSR	HEXOUT	DISPLAY RESULTS
00300	TFR	B,A	PUT B INTO A
00310	BSR	HEXOUT	DISPLAY RESULTS
00320	BRA	OUT	FINISHED
00330 HEXOUT	PSHS	A	SAVE CONTENTS
00340	LSRA		TURN BYTE INTO
00350	LSRA		NIBBLE
00360	LSRA		" "
00370	LSRA		" "
00380	BSR	OUTA1	SCREEN CORRECT
00390	PULSA		RENEW CONTENTS
00400	ANDA	#\$0F	STRIP MSN
00410 OUTA1	CMPA	#\$09	9 OR MORE?
00420	BGT	AF	MUST BE A-F
00430	ADDA	#\$70	MUST BE 9 OR <
00440	BRA	OX	SKIP ALPHA
00450 AF	ADDA	#\$37	ALPHA OFFSET
00460 OX	STA	,X+	PRINT NUMBER
00470	RTS		
00480 OUT	PULS	X,D	RESTORE REGS
00490 RETURN	FCB	\$7E	JUMP OPCODE
00500	FDB	0	
00510	END	START	

QUESTIONNAIRE FOR LESSON ONE

Q1: What is the CHR character that is beside the flashing cursor when in the command mode?

A1?

Q2: What is the command to input the listing?

A2?

Q3: How would you start inputting the listing at line 500?

A3?

Q4: Where do you find the REM lines in the listing?

A4?

Q5: What is the data called that is entered in column three?

A5?

Q6: What is the data called that is entered in column four?

A6?

Q7: What is the data called that is entered in column five?

A7?

Q8: How do we list the whole program?

A8?

Q9: What command is used to correct line 170?

A9?

Q10: How would we save the listing under the name COCO so that we could call it back to change or add to the listing at a later date?

A10?

Q11: How would we save the listing so that we can 'CLOADM"COCO":EXEC'?

A11?

Q12: How do we read in the source code after it has been saved?

A12?

Q13: How would we step through the listing one line at a time?

A13?

Q14: What are four different commands used while in the editing mode?

A14?

ANSWERS FOR LESSON ONE

A1: The asterisk.

A2: I for input.

A3: I500 for INPUT starting at line 500/

A4: Immediately following the asterisk or in column five.

A5: The opcode.

A6: The operand.

A7: The comment lines.

A8: By typing 'P#:'.

A9: By typing 'E170'.

A10: By typing 'W COCO' to write in ASCII.

A11: By typing 'A COCO' as in assemble to machine language.

A12: By typing 'L <filename>' as in load the source listing.

A13: With the down arrow key.

A14: 'C' = change,

'I' = input,

'A' = start again,

'D' = delete,

'H' = hack,

'K' = kill -

HINT....

List Disable

If you don't want anyone else to see your program, type in POKE383,57. This will return garbage whenever you type in LIST <ENTER>. To disable this, type in POKE383,126.

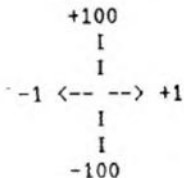
JOYMEM

by Justin Lipton UTILITY 16K + Joystick

JOYMEM is a program where, using the joystick you can go through the computers RAM. Some of the interesting things you will see are Color Basic Words, Messages, errors, Extended Basic and Disk Basic facts also.

You will also see where this program starts, finishes, and what it consists of in the computers memory.

You can go through the memory by following this diagram.



By pressing the fire button you can go through the memory very quickly.

The Listing:

```

0 GOTO10
1 '***** JOYMEM *****
2 '**** JUSTIN LIPTON *****
3 SAVE"109:3":END
10 CLS
20 P=0:A=32
30 H=JOYSTK(0):V=JOYSTK(1):R=P
40 IF H=63 THEN P=P+1
50 IF H=0 THEN P=P-1
60 IF V=0 THEN P=P+100:CLS
70 IF V= 63 THEN P=P-100:CLS
80 IF R=P THEN 30
90 IF P<1 THEN P=1
100 IF PEEK(65280)=126 OR PEEK(65280)=254 THEN P=P+13107:CLS
110 IF P>65535 THEN P=65535
120 PRINT @ 20, "LOC:"P
130 L=PEEK(P)
140 IF L=0 THEN L=32
150 PRINT @ A, CHR$(L);:A=A+1
160 IF A>510 THEN A=32
170 GOTO 30

```

LOWERCASE

by Graham Pollock
16K CB UTILITY

IHAVE HERE a small demo program for the latest CoCo 2. It will give you true lower case screen on this computer. No hardware modifications needed. It's gotta be the one with the 3 coloured stripes on the badge though! Bye for now!

The Listing:

```

1 REM Lower case by Grahame Pollock, 24 KENT St, Minto, 2566
2 GOTO10
3 SAVE"108:3":END
10 PRINT"POKES supplied by Mike Fahy":PRINT"on the CoCo page of":PRINT"The MC-10 User Group News letter":PRINT"Box 103,":PRINT"Owensville IN47665,":PRINT"U.S.A."
20 PRINT
30 PRINT"This program shows the lower case screens for the CoCo2 with the stripes on the case":PRINT:PRINT"Press any key for the next screen"
40 POKE359,13
50 GOSUB140
60 POKE65314,17
70 GOSUB140
80 POKE65314,81
90 GOSUB140
100 POKE65314,48
110 GOSUB140
120 POKE65314,89
130 GOSUB140:POKE65314,5:GOTO50
140 IS=INKEY$:IFIS="" THEN140
150 RETURN

```

```

223 PRINT" * COPYRIGHT 1986 BY ERIC WHITE *";
224 PRINT" * ALL RIGHTS RESERVED *";
225 PRINT"*****";
226 PRINT:PRINT" AFTER PROGRAM STARTS "
227 PRINT:PRINT" PRESS THE SPACEBAR "
228 PRINT" TO SELECT THE KEY BOARD "
229 PRINT" OR
230 PRINT" PRESS THE FIRE BUTTON "
231 PRINT" TO SELECT THE JOYSTICK "
232 PRINT:PRINT" FOR INPUT DEVICE "
233 IF X<>99 THEN GOTO 69 ELSE RETURN

```

CORRECTION

Last month in Australian Rainbow, we inadvertently omitted the following lines from "CoCo-Nect-a-Dot" (page 52).

The Listing:

```

173 DATA 12,176,AS COLORFUL AS A
174 DATA 96,68,L,72,84,R,68,68,L,128,58,R,128,68,R,56,56,L,68,36,L,128,84,R,108,92,L,92,38,L,116,32,R,72,104,R,44,92,L,36,76,L,68,84,L,88,148,L,28,148,L,52,168,L,148,168,R,188,148,R,88,148,R,88,88,END
175 '
176 DIM P(13),M(13),S(13),A$(26),N$(25),K(88),L(18)
177 A=54:B=15:OY=45:OX=216:KJ=1:X=OX:Y=OY
178 '
179 ' CHARACTER DATA
180 '
181 D$="EFGHURB3" 'DOT
182 P$="U5E15F5G3H3G11BF2E11H4E5R2F3D2G17L5E"
183 N$(8)="BRNR2HU2ER2FD2GBR3"
184 N$(1)="BEHR4BD2BR2"
185 N$(2)="U3EFD3R2NU4BR2"
186 N$(3)="U3EFND2EFD3BR2"
187 N$(4)="BU3NR4F3NU4BR3"
188 N$(5)="NU4R2U3EFD3BR2"
189 N$(6)="BU3BNED2FR2HU2EFD2GBR3"
190 N$(7)="U4RF3BD2BR2"
191 N$(8)="BRHU2EFND2EFD2GHGBR5"
192 N$(9)="BRHU2ENR2FD2GBR2BEU2

```

```

HB"
193 A$(1)="BRNR3HU2ERND4R2BD4BR2"
194 A$(2)="NR4U3EFND3EFD3BR2"
195 A$(3)="BRNR2HU2EBR2FD2GBR3"
196 A$(4)="NR4U3ER2FD3BR2"
197 A$(5)="NU3R2NU2R2NU4BR2"
198 A$(6)="NU4R2NU2R2BR2"
199 A$(7)="NR4U4BR2NDRFD3BR2"
200 A$(8)="R2NR2U4NL2R2BD4BR2"
201 A$(9)="BU2R4BD2BR2"
202 A$(10)="BU4R3FD2GLBR4"
203 A$(11)="NR4BU4F2ND2E2BD4BR2"
204 A$(12)="R4NU4BR2"
205 A$(13)="NR4E2H2R4BD4BR2"
206 A$(14)="NR4E4NL4BD4BR2"
207 A$(15)="BRNR2HU2ER2FD2GBR3"
208 A$(16)="NR4U3EFD3BR4"
209 A$(17)="BRNR2HU2ER2FDNLRDGBR3"
210 A$(18)="NR4U3EFND3BDE2BD4BR2"
211 A$(19)="BU3NBED2FEU2EFD3BGBR3"
212 A$(20)="U2NU2R4BD2BR2"
213 A$(21)="NR3BU4R3FD2GBR3"
214 A$(22)="NR2BU4R2F2G2BR4"
215 A$(23)="NR4BU4R4G2F2BR2"
216 A$(24)="E4BL4F4BR2"
217 A$(25)="BU4F2NG2R2BD2BR2"
218 A$(26)="U4F4NU4BR2"
219 '
220 FORX=1TO16:PRINTSTRING$(32,32);:NEXTX
221 PRINT"*****";
222 PRINT" * BASIC COCO-NECT A DOT V:1.8 *";

```

Spotlight

THE WINNERS

People keep asking me, "where is all the CoCo 3 software?"

Well, after this competition, I can tell you!!

It is right here in this magazine - and in the disks and tapes that will come from it!

We've been inundated with excellent quality programs for the CoCo 3!

I might add as an aside, that this magazine won't normally be quite so full of CoCo 3 progs - but with the quantity of "3" programs we have here, we thought the "3" owners might like to get a few progs in one hit to run on their new machines!

I'm only mentioning one name so far as winners in this competition go, because his program, "Artist", is so good!

But other entrants take heart! The next competition deadline is 31st July, 1987 - there is a lot of time yet to catch up and surpass the efforts of Craig Stewart - the author of "Artist", a program which appears this month and which is an EXCELLENT graphics creation utility.

"ARTIST" uses the 16 colour screen to make beautiful pictures which can be saved to disk or tape.

CoCoOz this month has two files from this program which

show the magnificent effects you can create!

Congratulations again to Craig, who wins a selection of Tandy CoCo products including a set of Joysticks, a Koala Pad and a box of disks - prizes to the value of about \$200.00!

And thanks go to Tandy for these prizes, their participation on this occasion has certainly proved valuable!

The Music Competition was supposed to be smaller again in scope!

But it blew out because we received music in Basic, in Music + format, in Musica format, and in Orchestra 90 format.

Tricky to judge of course, so we opted for the coward's way out and decided to award prizes in all four categories!!

Again we received music from many people. Thank you to all who entered the competition - please try again!

In the Basic section, Johanna Vagg and family won the prize with their clever use of the PLAY statement to produce some interesting effects.

In fact Craig Stewart discovered one of the same effects, but the family from

Forbes took the experiment further!

Johanna's Basic music winner is "Laura".

In the Music + section, Johanna Vagg also won with her rendition of "Ave Maria".

Johanna specialises in using this program so we weren't surprised in the least when she won this section of the competition.

Harvey Smith won the Musica section with his rendition of "Stardust" by Hoagy Carmichael.

Musica is available from Computer Hut Software and is a very useful program to have if you enjoy creating music on your computer.

Finally, Michael Monck won the Orchestra 90 section with two pieces, "Axel" and "Popcorn".

Orchestra 90 is a ROM pack program you can purchase from Tandy which provides Organ quality stereo music, and which plugs into the ROM port of your CoCo.

Michael's two music files show this ROM pack off to considerable advantage.

The winners of each of the sections of the Music Competition all receive two boxes of our DSDD disks.

Congratulations

ARTIST



GRAPHICS UTILITY
CoCo 3 disk based System

by Craig Stewart

THIS IS MY entry to the CoCo 3 competition. It is a graphics drawing and editing program that takes full advantage of the CoCo 3's new graphics screen (320 x 192 16 colour).

Although the main part of the drawing program took only a little while to write (being modular, in a piece by piece format and written in Basic) the hardest part was developing a routine to save the 32K graphics screen to tape or disk. (A 32K that doesn't really exist to the processor).

At length I found a way using a bit of basic and M/L by opening and closing files. This only takes about fifteen seconds to disk but a lot longer to tape.

These two routines (save and load) may be handy for other programs needing their graphics screens saved. About the program...

The program is along the same sort of lines as some other basic graphic drawers for the previous CoCo's that were in the Rainbow magazine except this one uses a more visual, user friendly approach.

The major features are:-

- * 1 to 6 modes of cursor control
- * Easy control over sixteen colours and sixty four palettes while viewing screen results.
- * Nine 'draw' string variable holders with automatic scale and angle options
- * A 'get and slide' copying feature
- * Borders
- * Save and load screen from disk or cassette - with full disk handling (dir and kill)
- * A write to screen
- * Flicker mode for a 'more interesting screen'
- * Automatic paint
- * Plus normal graphic stuff like circles, lines and boxes etc etc.

They can all be controlled by just the keyboard if desired (joystick optional).

COCO 3

COMPETITION

WINNER

OPERATION

1-6 : -cursor control mode:
1=keyboard single 2= keyboard continuous (1 and 2 require cursor keys) 3= joystick slow 4= joystick fast 5= joystick vector 6= joystick direct

BREAK :- exit program (prompted y/n)

B :- brush size.. choose 1-4 (small 1-4 large)

C :- colour select.. place flashing box over desired colour using left/right arrow keys or joystick. Press button to select colour or up/down arrow keys or joystick to select palette.

D :- draw string.. answer prompt.. choose 'E' to edit string, 1-9 to draw string, 'S' to set scale, 'A' to set angle.

E :- edge (border).. create a border around a picture (square or round.. answer prompt).

F :- flash colour.. using this it is possible to flash colours periodically to the screen to create flashing titles, limited animation etc. (You will have to have your wits about you to use this effectively.)

G :- get and slide.. for duplicating images.. position code in edit mode to one corner of 'get' rectangle.. press 'G'.. position cursor in other corner.. press button.. move about screen to new position.. press button to 'stamp' image there. (pressing 'O'-OR , or 'P'-PSET will change logical operators for different effects-experiment!) Press 'Q' when finished.

H :- hop factor.. cursor movement factor 1-5

I :- circle.. position cursor at centre in edit mode... press 'I' .. move to edge of circle.. press button- circle will appear (pressing 'R' in this mode will allow you to change the h/w ratio).

J :- join.. will join gaps between brush in edit mode (toggle on/off)

K :- kill disk file

L :- load disk/cassette picture

M :- menu

P :- paint (automatic).. this will scan the area directly

NOTES

* This program was written on a colour television set, and works fine (although graphics may be a little fuzzy)

* The forty column text has been moved over two spaces for those with the cut-off text problem

* The spacebar doubles as a joystick button

* Most modes can be exited back to the screen editor by pressing 'Q'uit

* All drawing and painting is done in the current colour

* Most commands are already 'prompted'

* A palette of sixteen colours has already been chosen; these can be changed before, during and after your picture has been created.

* For the sake of clarity, please leave the first colour in the colour selection black, and the last colour white.

* Finally, the cursor control mode, and the 'H'op factor can both be changed in any cursor using mode.

above it for a border colour.. the first colour it contacts other than the colour the cursor is currently on, it will treat as a border colour, and start painting (be very careful when painting over parts of the picture to get border colour directly above the cursor).

Q :- will 'Q'uit most modes and return to editor

R :- report.. will report the current status conditions of the program (cursor position, colour palette, what's in the draw strings, and lots of other stuff).

S :- save picture to cassette or disk- this also saves your draw strings, flash information, and your particular colour palette to disk.

W :- write to screen.. position the cursor in edit mode and press 'W'. Start writing from keyboard, and press 'ENTER' when finished.

X :- line/box/box fill.. position cursor on one corner in edit mode.. press 'X' .. position cursor at opposite corner and press 'L' for a line, 'B' for a box or 'F' for a full box.

Z :- disk directory.
<SHIFT><CLEAR> :- CLS

At the end of the COCOOZ tape or disk you will find a demo picture ready to load in.

I urge anyone using this program to experiment to obtain the full potential of it. Lots of special effects can be obtained using some of the modes. The program is fully break and error trapped so you cannot crash the program (although you can destroy your picture.. eg. 'paint').

The Listing:

```

0 GOTO10
3 SAVE"181:3":END
10 'COCO-3 ARTIST
20 'BY CRAIG STEWART
30 GOTO 50
40 SAVE "ARTIST":STOP
50 POKE65497,0
60 PALETTE CMP:WIDTH40:CLS1:LOCA
TE14,5:ATTR 2,4,B,U:PRINT"COCO-3
ARTIST";:ATTR2,0:PRINT:LOCATE 8
,15:PRINT"written by Craig Stewa
rt":LOCATE 13,17:PRINT"for the C
OCO-3"
70 POKE65496,0:PLAY"V31T25501":F
OR X=1 TO 30:PLAY"AFCEADBEAEDV-"
:NEXT X:POKE65497,0
80 CLEAR 4000:DIM B(17),C(17),D$(
9),N$(9):FOR X=0 TO 15:READ C:B
(X)=C:NEXT X
90 DATA 0,1,9,15,29,30,16,22,32,
36,38,37,43,48,54,63
100 GOSUB 2610
110 ON BRK GOTO 2730:ON ERR GOTO
2760
120 JS=1:SC=4:AN=0:RA=1
130 P1$="V3105T20CV10CV3C"
140 P2$="V3101T20CV10CV3C"
150 P3$="T50V3104C":PQ$="T50V310
2C"
160 HBUFF 1,1800:HBUFF 2,1300:HB
UFF 3,2400:HBUFF 4,2400
170 FORX=0 TO 15:PALETTE X,0:NEXT
:HSCREEN2:HCLS0:C=0:FOR X=10 TO
310 STEP 19:HCOLOR C:HLINE(X,91)
-(X+14,99),PSET,BF:C=C+1:NEXT X:
HGET(0,90)-(320,100),1
180 HCLS0:HCOLOR 15:HPRINT(2,2),
"BRUSH SIZE":HPRINT(2,3),1 2
3 4":HSET(20,35,5):HLINE(42,34)
-(44,36),PSET,BF:HLINE(64,33)-(6
8,37),PSET,BF:HLINE(88,32)-(94,3
8),PSET,BF:HLINE(15,12)-(96,40),
PSET,B:HGET(15,12)-(96,40),2
190 HCLS0:GOSUB340:X0=160:Y0=96
200 GOSUB 220:IF BUTTON(0)=1 ORP
EEK(345)<>255 THEN HCOLOR CO:HLI
NE(X0-BR,Y0-BR)-(X0+BR,Y0+BR),PS
ET,BF
210 GOTO 200
220 X1=X0:Y1=Y0:P=HPOINT(X0,Y0):
X3=X0:Y3=Y0:HSET(X0,Y0,15)
230 GOSUB 370
240 IF (X1-BR)<0 THENX1=BR ELSE
IF (X1+BR)>319 THEN X1=319-BR
250 IF (Y1-BR)<0 THENY1=BR ELSE

```

```

IF (Y1+BR)>191 THEN Y1=191-BR
260 IF JO=1 THEN IFBUTTON(0)=1 O
R PEEK(345)<>255 THEN HLINE(X0,Y
0)-(X1,Y1),PSET
270 X0=X1:Y0=Y1
280 HSET(X3,Y3,0):GOSUB660
290 HSET(X3,Y3,P):RETURN
300 'HSCREEN WITHOUT PCLS
310 POKE&HE6DC,57:HSCREEN2:POKE&
HE6DC,141:GOTO 340
320 PALETTE CMP:WIDTH 40:CLS1:L=
LEN(A$):LOCATE (40-L)/2,1:ATTR2,
4,B,U:PRINT A$;:ATTR2,0:PRINT
330 RETURN
340 FOR X=0 TO 15:PALETTE X,B(X)
:NEXT:RETURN
350 HCOLOR15:HGET(0,181)-(319,19
1),4:HPRINT(2,23),A$:RETURN
360 HPUT(0,181)-(319,191),4,PSET
:RETURN
370 'TYPE OF MOVEMENT ROUTINE
380 JX=JOYSTK(0):JY=JOYSTK(1)
390 ON MO GOTO 410,440,500,590,6
10,640
400 'KEYBOARD STATIC
410 IF A$="" THEN 420 ELSE IF A$
="" THEN Y1=Y1-JS ELSE IF A$=C
HRS(8) THEN X1=X1-JS ELSE IF A$=C
HRS(9) THEN X1=X1+JS ELSE IF A$=
CHRS(10) THEN Y1=Y1+JS
420 RETURN
430 'KEYBOARD CONTINUOUS
440 IF PEEK(343)<>255 THEN MX=-1
450 IF PEEK(344)<>255 THEN MX=1
460 IF PEEK(341)<>255 THEN MY=-1
470 IF PEEK(342)<>255 THEN MY=1
480 X1=X1+MX*JS:Y1=Y1+MY*JS:MX=0
:MY=0
490 RETURN
500 'JOYSTK STATIC
510 IF NM=0 THEN NM=1:GOTO 570
520 NM=0
530 IF JX=63 THEN X1=X1+JS
540 IF JY=0 THEN Y1=Y1-JS
550 IF JY=63 THEN Y1=Y1+JS
560 IF JX=0 THEN X1=X1-JS
570 RETURN
580 'JOYSTK CONTINUOUS
590 GOTOX 520
600 'JOYSTK VECTOR
610 X1=X1+INT((JX-31.5)*JS*.1):Y
1=Y1+INT((JY-31.5)*JS*.1)
620 RETURN
630 'JOYSTK ABSOLUTE
640 X1=JX*5.1:Y1=JY*3.1
650 RETURN
660 'KEYBOARD POLL

```

```

670 A$=INKEYS:IF A$="" THEN RETU
RN
680 IF A$=CHRS(8) OR A$=CHRS(9)
OR A$=CHRS(10) OR A$="" OR A$=""
THEN RETURN
690 HSET(X3,Y3,P)
700 IF A$="1" AND A$<"7" THEN M
O=VAL(A$):PLAYPI$:RETURN
710 IF A$="H" THEN PLAYPI$:GOTO
2690
720 IF KN=1 THEN RETURN
730 IF A$="P" THEN PLAYPI$:GOTO9
20
740 IF A$="C" THEN PLAYPI$:GOTO1
010
750 IF A$="X" THEN PLAYPI$:GOTO1
120
760 IF A$="B" THEN PLAYPI$:GOTO1
240
770 IF A$="E" THEN PLAYPI$:GOTO1
270
780 IF A$=CHRS(92) THEN PLAYPI$:
GOTO1370
790 IF A$="M" THEN PLAYPI$:GOTO1
380
800 IF A$="R" THEN PLAYPI$:GOTO2
020
810 IF A$="S" THEN PLAYPI$:GOTO1
570
820 IF A$="L" THEN PLAYPI$:GOTO1
730
830 IF A$="I" THEN PLAYPI$:GOTO1
890
840 IF A$="D" THEN PLAYPI$:GOTO1
430
850 IF A$="G" THEN PLAYPI$:GOTO2
140
860 IF A$="J" THEN PLAYPI$:GOTO2
290
870 IF A$="W" THEN PLAYPI$:GOTO2
320
880 IF A$="F" THEN PLAYPI$:GOTO2
420
890 IF A$="Z" THEN PLAYPI$:GOTO1
840
900 IF A$="K" THEN PLAYPI$:GOTO1
870
910 RETURN
920 'PAINT
930 HSET(X0,Y0,P)
940 P=HPOINT(X0,Y0)
950 FOR X=Y0 TO 0 STEP -1:Z=HPOI
NT(X0,X):IF Z<>P THEN 970
960 NEXT X
970 HPAINT(X0,Y0),CO,Z
980 P=CO
990 RETURN
1000 'COLOR CHART

```

Tandy
ELECTRONICS

MODEL 4 TANDY DESKTOP COMPUTER



**SAVE
\$700**

Reg 1999.00 **\$1299**

What a price breakthrough! The Tandy Model 4 is the perfect computer for busy managers, educators and home computer enthusiasts. Has a spacious 64K memory expandable to an enormous 128K! Featuring a numeric keypad, dual disk drives, parallel

printer interface and a huge ready-to-run software library. Includes an easy to follow owners manual and introductory booklet to help get you started. With amazing value like this you wouldn't want to miss out! 26-1069

Tandy Computer Cassette Recorders



A
89⁹⁵

B
59⁹⁵

A. CCR-81. Ideal for loading and saving your own programs and data on cassette tapes. Requires 4 "C" batteries (not incl.) or 240V AC. Includes cable. 26-1208
B. CCR-82. Our *lowest-priced* computer cassette recorder. Includes cable. Requires 4 "AA" batteries or AC adapter. (Batteries & AC adapter not incl.) 26-1209

Fast Daisy Wheel Printer



Reg 999.00

\$699 SAVE \$300

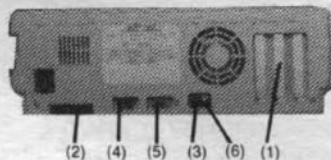
DWP-220. Here's an extremely affordable way to expand your computer! Prints over 200 wpm in clear, sharp, "electric typewriter" characters. Uses interchangeable print wheels that simply drop in — no mess. 26-1278



SAVE \$800

Reg 1999.00

\$1199



SPECIFICATIONS

Microprocessor: Intel 8088 processor. Clock speed, 4.77 MHz. Operating System: Includes Microsoft's advanced MS-DOS 2.11 with BASIC. **Memory:** 128K RAM, expandable to 640K. **Keyboard:** 90-key sculptured, including numeric keypad. Twelve programmable Special Function keys. **Video Display:** Optional high-resolution, non-glare 30.48cm monochrome (green) or color monitor. 80 or 40 characters per line by 25 lines. High-resolution monochrome and color graphics (displaying 8 colors selected from 16). **Disk Drives:** One built-in double-sided, double-density, 360K (formatted) thin-line 13-34cm mini-floppy, 48 tracks per inch. **Internal Expansion:** (1) Three user-accessible IBM PC-compatible 25cm card slots, second 360K Floppy Disk Drive. **External Connections:** (2) Standard parallel printer port, (3) monochrome monitor, (4) Light pen adapter, two joysticks, (5) RGBI Color Monitor, (6) composite video and audio. **Power:** 240 VAC, 50Hz.

Excellence in PC Business Computing

The Tandy 1000 is indicative of our pledge to making high quality affordable computers. Other PC compatibles don't offer you adapters for your peripheral devices or include a software program that may be all you need! Tandy 1000 is the basis for greater business efficiency.

25-1000

Tandy 1000 with Monitor and Deskmate Software

SAVE **\$848⁰⁰** **1499**

Reg 2347.00

An offer that's unique since you'll probably never need to add to the Deskmate Software we've included here. It's a word processor, spreadsheet, inventory list and has a menu that greets you each morning with a calendar and appointment schedule. With a modem you'll be able to converse with other computers. The 30cm monochrome monitor is non-interlaced for sharp clear display.

25-1000 3211.

- **The Complete Computer Package That Includes DESKMATE Software as a BONUS**
- **Also Includes A Monochrome Monitor That's Phosphor Green For Easy On The Eyes Composing.**

WE SERVICE WHAT WE SELL!

**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

**Tandy
ELECTRONICS**

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

**Nearly
350 Stores
Australia-
Wide**

```

1010 HGET(0,181)-(319,191),4:HPU
T (0,181)-(319,191),1,PSET
1020 JX=JOYSTK(0):JY=JOYSTK(1):A
S=INKEYS:IF JX=21 AND JY=0 THENJ
X=1:JY=1
1030 IF JX=0 OR AS=CHRS(8) THEN
K=K-1:PLAYPPS
1040 IF JX=63 OR AS=CHRS(9) THEN
K=K+1:PLAYPQS
1050 IF K<0 THEN K=0 ELSE IF K>1
5 THEN K=15
1060 HCOLOR 15,0:HLINE(K*19+8,18
1)-(K*19+26,191),PSET,B:HLINE(K*
19+8,181)-(K*19+26,191),PSET,B
1070 IF JY=0 OR AS="" THEN B(K)
=B(K)+1:PLAY PPS:IF B(K)>63 THEN
B(K)=0
1080 IF JY=63 OR AS=CHRS(10) THE
N B(K)=B(K)-1:PLAYPQS:IF B(K)<0
THEN B(K)=63
1090 PALETTE K,B(K)
1100 IF BUTTON(0)=1 OR AS="" TH
EN CO=K:PLAY PIS:HPUT(0,181)-(31
9,191),4,PSET:RETURN
1110 GOTO 1020
1120 'LINE,BOX ETC
1130 KN=1
1140 X1=X0:Y1=Y0:X4=X0:Y4=Y0
1150 HSET(X0,Y0,CO):HCOLOR CO
1160 GOSUB 220
1170 IF AS="" THEN 1160
1180 IF AS="L" THEN HLINE(X4,Y4)
-(X1,Y1),PSET:GOTO 1230
1190 IF AS="B" THEN HLINE(X4,Y4)
-(X1,Y1),PSET,B:GOTO 1230
1200 IF AS="F" THEN HLINE(X4,Y4)
-(X1,Y1),PSET,BF:GOTO 1230
1210 IF AS="Q" THENPLAYPPS:GOTO1
230
1220 GOTO 1160
1230 P=CO:KN=0:RETURN
1240 HGET(15,12)-(96,40),3:HPUT(
15,12)-(96,40),2,PSET
1250 AS=INKEYS:IF AS="" THEN 125
0 ELSE IF AS<"1" OR AS>"4" THEN
1260 ELSE BR=VAL(AS)-1
1260 HPUT(15,12)-(96,40),3,PSET:
RETURN
1270 'BORDER
1280 CLS2:AS="EDGE (BORDER) AROU
ND THE PICTURE":GOSUB320
1290 LOCATE 4,10:PRINT"what type
of border do you want":LOCATE 6
,12:PRINT"1. SQUARE":LOCATE6,13:
PRINT"2. ROUND":LOCATE 6,14:PRIN
T"3. quit"
1300 AS=INKEYS:IF AS="3" THEN 30
0 ELSE IF AS="" OR AS<"1" OR AS>
"2" THEN 1300
1310 LOCATE 4,17:PRINT"Border si
ze? (pixels from edge)":LINE INP
UTPS:PI=VAL(PX)
1320 IF PI<2 THEN LOCATE4,19:PRI
NT"TOO SMALL!":FOR X=1 TO 1000:N
EXT:LOCATE4,19:PRINT:GOTO 1310
1330 GOSUB 310:GOSUB 340
1340 IF AS="1" OR AS="3" THEN HC
OLOR CO:HLINE(PI,0)-(0,191),PSET
,BF:HLINE(0,PI)-(319,0),PSET,BF:
HLINE(319-PI,0)-(319,191),PSET,B
F:HLINE(0,191-PI)-(319,191),PSET
,BF
1350 IF AS="2" THEN HCIRCLE(160,
96),160-PI,CO,192/320:HPAINT(0,0
),CO,CO
1360 RETURN

```

```

1370 HCLS CO:P=CO:RETURN
1380 AS="MAIN MENU":GOSUB320
1390 PRINT:PRINT" P - PAINT":PR
INT" C - COLOUR SELECTION":PRIN
T" X - LINE (l-line, b-box, f-f
ill box)":PRINT" B - BRUSH SIZE
(1-4)":PRINT" E - EDGE (BORDER
)":PRINT" M - MENU":PRINT" R -
REPORT ON CURRENT CONDITIONS"
1400 PRINT" S - SAVE PICTURE TO
TAPE OR DISK":PRINT" L - LOAD
PICTURE FROM TAPE OR DISK":PRINT
" I - CIRCLE (r- set height/wid
th)":PRINT" D - DRAW STRING ":P
RINT" G - GET AND MOVE FIGURE":
PRINT" J - JOIN BETWEEN POINTS
(TOGGLE)"
1410 PRINT" H - SET HOP (MOVE)
FACTOR":PRINT" <SHFT> CLEAR - C
LEAR SCREEN":PRINT" 1-6 : MOVEM
ENT MODE":PRINT" K - KILL DISK
FILE":PRINT" Z - DISK DIR":PRIN
T" F - FLICKER MODE":PRINT" W
- WRITE"
1420 GOTO 2130
1430 'DRAW
1440 AS="1-9 (FIG1-9), E-EDIT, A
-ANGLE,S-SCALE":GOSUB350
1450 HSET(X0,Y0,CO)
1460 AS=INKEYS:IF AS="" THEN 146
0
1470 GOSUB360
1480 BS=AS:IF AS>"1" AND AS<"9
" THEN AS="B"+STR$(X0)+" "+STR$(
Y0)+"C"+STR$(CO)+"S"+STR$(SC)+"
A"+STR$(AN)+D$(VAL(AS)):HDRAW AS
:P=CO:RETURN
1490 IF AS="A" THEN AS="DRAW ANG
LE":GOSUB320:PRINT:PRINT" WHAT
IS NEW DRAW ANGLE (0-3)":INPUT
AN:IF AN<0 OR AN>3 THEN PLAY PQ$
:AS="A":GOTO 1490 ELSE GOSUB310:
RETURN
1500 IF AS="S" THEN AS="DRAW SCA
LE":GOSUB320:PRINT:PRINT" What
is new draw scale (1-63)":INPUT
SC:IF SC<1 OR SC>63 THEN PLAYPQ
$:AS="S":GOTO 1500 ELSE GOSUB310
:RETURN
1510 IF AS<"E" THEN RETURN
1520 AS="EDIT DRAW STRING":GOSUB
320:PRINT:PRINT" Which draw st
ring to edit (1-9)":
1530 AS=INKEYS:IFAS<"1" OR AS>"9
" THEN 1530 ELSE X=VAL(AS)
1540 PRINT:PRINT" Old draw str
ing no.":X;X" contains":PRINT:ATT
R 2,4:IF D$(X)="" THEN PRINT"
...nothing...":ELSE PRINT"
":D$(X);"";
1550 ATTR 2,0:PRINT:PRINT:PRINT
" Input new draw string ('Q' t
o abort)":PRINT:ATTR 2,4:PRINT"
":LINEINPUT AS:IFAS="Q" THEN1
560 ELSE D$(X)=AS:PRINT:INPUT"Na
me of string? (1-8 char)":NS:NS(
X)=LEFT$(NS,8)
1560 P=CO:GOSUB310:RETURN
1570 'SAVE SCREEN
1580 POKE65496,0
1590 AS="SAVE GRAPHICS SCREEN":G
OSUB320
1600 PRINT:PRINT" SAVE TO (C)A
SSETTE, (D)ISK, (Q)UIT ?"
1610 BS=INKEYS:IFBS="Q" THEN 172
0 ELSE IFBS<"C" AND BS<"D" THE

```

```

N1610
1620 GOSUB1630:GOTO 1680
1630 PRINT:PRINT" What is name
of picture? (1-8 let.)":PRINT:P
RINT" ";ATTR2,4:LINEINPUTAS:A
S=LEFT$(AS,8):AS=AS+STRING$(8-LE
N(AS),32):ATTR 2,0:PRINT
1640 IFBS="C" THEN 1670 ELSE PRI
NT:PRINT" Which drive ? (0-3)"
1650 CS=INKEYS:IFCS<"0" OR CS>"3
" THEN 1650
1660 DRIVE VAL(CS):PRINT
1670 PRINT" please be patient.
..":RETURN
1680 IFBS="C" THEN S=-1 ELSE S=1
:AS=AS+"/PCT"
1690 OPEN "O",#S,AS:IF S=-1 THEN
POKE&H6F,255 ELSE POKE&H6F,1
1700 EXEC&HE00:FORX=1 TO 9:PRINT
#S,D$(X):PRINT#S,N$(X):NEXT X:FO
R X=0 TO 15:PRINT#S,B(X):PRINT#S
,C(X):NEXT:CLOSE
1710 PRINT:PRINT"...FINISHED...
..":PLAYPIS:FORX=1TO1000:NEXT
1720 POKE65497,0:GOSUB310:RETURN
1730 'LOAD SCREEN
1740 POKE65496,0
1750 AS="LOAD GRAPHICS SCREEN":G
OSUB320
1760 PRINT:PRINT" LOAD FROM (C
)ASSETTE, (D)ISK, (Q)UIT?"
1770 BS=INKEYS:IF BS="Q" THEN 17
20 ELSE IFBS<"C"AND BS<"D" THE
N 1770
1780 GOSUB1630
1790 IF BS="C" THEN S=-1 ELSE S=
1:AS=AS+"/PCT"
1800 OPEN "I",#S,AS:IFS=-1 THENP
OKE&H6F,255 ELSE POKE&H6F,1
1810 EXEC&HF00:FORX=1TO9:INPUT#S
,D$(X):INPUT#S,N$(X):NEXTX:FOR X
=0 TO 15:INPUT#S,B(X):INPUT#S,C(
X):NEXTX:CLOSE
1820 GOTO 1710
1830 'DISK DIR
1840 AS="DISK DIR":GOSUB320:PRIN
T:PRINT" WHICH DRIVE ? (0-3)"
1850 AS=INKEYS:IFAS="Q" THEN 310
ELSE IFAS<"0" OR AS>"3" THEN 18
50 ELSE DRIVE VAL(AS):POKE65496,
0:DIR:POKE65497,0:PRINT:PRINT"
FREE =":FREE(VAL(AS)):" GRANULE
S":PRINT:PRINT" PICTURES HAVE
EXTENSION 'PCT':GOTO 2130
1860 'KILL FILE
1870 AS="KILL FILE":GOSUB320:POK
E65496,0:GOSUB1630:IFAS="Q
" THEN 1880 ELSE BS=AS+"/PCT":K
ILL BS:PRINT:PRINT" ";AS;" KIL
LED"1880 POKE65497,0:FORX=1 TO 1
000:GOTO310
1890 'CIRCLE
1900 KN=1:X4=X0:Y4=Y0:PI=P
1910 HSET(X0,Y0,CO)
1920 GOSUB 220
1930 IF AS="Q" THEN PLAYPPS:GOTO
1980
1940 IF AS="R" THEN GOSUB1990
1950 IF BUTTON(0)=0 ANDAS<" "
THEN1920
1960 D=INT(SQR((X4-X1)^2+(Y4-Y1)
^2)+.5)
1970 HCIRCLE(X4,Y4),D,CO,RA
1980 X0=X4:Y0=Y4:Y3=Y4:X3=X4:KN=
0:P=PI:RETURN
1990 AS="CIRCLE H/W RATIO":GOSUB

```

```

320
2000 PRINT:INPUT" WHAT IS NEW H
/V RATIO";RA
2010 GOSUB310:RETURN
2020 'REPORT
2030 AS="REPORT ON CURRENT CONDI
TIONS":GOSUB320:
2040 PRINT:PRINT" CURSOR POSIT
ION:HOR=";X1;" ,VERT=";Y1
2050 PRINT:PRINT" CURRENT COLO
UR=";CO;" PALETTE NO.;"B(CO)
2060 PRINT:PRINT" CURSOR MODE
=";MO
2070 PRINT" MOVE FACTOR =" ;JS
2080 PRINT" (J)OIN IS " ;:IF JO
=1 THENPRINT"ON" ELSE PRINT"NOT
ON"
2090 PRINT" (D)RAW SCALE =" ;SC
;" DRAW ANGLE =" ;AN
2100 PRINT" C(I)RCLE H/W RATIO
=" ;RA
2110 PRINT:PRINT" DRAW STRINGS
1-9 CONTAIN ..":FORX=1 TO9:IF D
$(X)=" THEN PRINT TAB(3)X;"noth
ing" ELSE PRINT TAB(3)X;N$(X)
2120 NEXT X
2130 IF INKEYS="" THEN 2130 ELSE
GOTO310
2140 'GET AND MOVE
2150 HSET(X1,Y1,CO):P1=P:KN=1:X1
=X0:Y1=Y0:X4=X0:Y4=Y0
2160 GOSUB 220
2170 IF AS="Q" THENPLAYPS$:GOTO2
280
2180 IF BUTTON(0)=1 OR AS=" " TH
EN 2190 ELSE 2160
2190 HSET(X4,Y4,P):X5=INT((X1-X4
)/2)*2:Y5=INT((Y1-Y4)/2)*2:BY=IN
T(ABS(X5)+1.5)*INT(ABS(Y5)+1.5)/
2:IF BY>2400 THEN PLAYPQ$:GOTO 2
160 ELSE HGET(X4,Y4)-(X4+X5,Y4+Y
5),3:X0=X4:Y0=Y4:PLAYPI$
2200 HGET(X4,Y4)-(X4+X5,Y4+Y5),4
:IF OP=0 THEN HPUT(X4,Y4)-(X4+X5
,Y4+Y5),3,PSET ELSE HPUT(X4,Y4)-
(X4+X5,Y4+Y5),3,OR
2210 IF AS="O" THEN OP=1:PLAYPI$
ELSE IFAS="P" THEN OP=0:PLAYPI$
2220 GOSUB220:IF BUTTON(0)=1 OR
AS=" " THEN 2270
2230 IF AS="Q" THEN PLAYPS$:HPUT
(X4,Y4)-(X4+X5,Y4+Y5),4,PSET:GOT
O 2280
2240 IF (X1+X5)>319 THENX1=319-X
5 ELSEIF (X1+X5)<0 THEN X1=-X5
2250 IF (Y1+Y5)>191 THEN Y1=191-
Y5 ELSE IF (Y1+Y5)<0 THEN Y1=-Y5
2260 HPUT(X4,Y4)-(X4+X5,Y4+Y5),4
,PSET:X4=X1:Y4=Y1:GOTO2200
2270 PLAYPI$:GOTO 2200
2280 KN=0:P=P1:RETURN
2290 'TOGGLE JOIN POINT ROUTINE
2300 IF JO=0 THEN JO=1 ELSE JO=0
2310 RETURN
2320 'WRITE TO SCREEN
2330 X=INT(X0/8):Y=INT(Y0/8)
2340 HCOLOR CO
2350 AS=INKEYS:IFAS="" THEN2350
2360 IF AS=>" " AND AS=<"z" AND
AS<>" " THEN2400
2370 IF AS=CHR$(13) THEN RETURN
2380 IF AS=CHR$(8)THENX=X-1 ELSE
IFAS=CHR$(9) THEN X=X+1 ELSE IF
AS=CHR$(10) THEN Y=Y+1 ELSE IF
AS=" " THEN Y=Y-1
2390 GOTO 2350

```

```

2400 IF X>39THEN X=0 ELSE IF Y>2
3 THEN Y=0 ELSE IF X<0 THEN X=-39
ELSE IF Y<0 THEN Y=23
2410 HPRINT(X,Y),AS:X=X+1:GOTO 2
350
2420 'FLICKER
2430 AS="Q-QUIT, E-EDIT, F-FLASH
,R-RATE":GOSUB350
2440 AS=INKEYS:IFAS=""THEN 2440
ELSE GOSUB360:IFAS="Q" THEN RETU
RN ELSE IF AS="E" THEN 2450 ELSE
IF AS="S" THEN 2530 ELSE IFAS="
R" THEN 2590 ELSE GOTO 2440
2450 AS="EDIT FLICKER":GOSUB320
2460 FOR X=0 TO 15
2470 LOCATE 4,X+5:PRINT"COLOUR "
;X;"=PAL";B(X);:LOCATE 4,3:PRINT
"FLICKER (Y/N)"
2480 AS=INKEYS:IFAS="N" THEN C(X
)=B(X):NEXT X:GOTO 2510 ELSE IF
AS<>"Y" THEN 2480
2490 LOCATE4,3:PRINT"WHAT IS NEW
COLOUR (0-63)?" ;:INPUTNC:IF NC<
0 OR NC>63 THEN 2490
2500 C(X)=NC:NEXT X
2510 GOTO 2590
2520 GOSUB310
2530 FOR X=0 TO15:PALETTE X,C(X)
:NEXT
2540 FOR X=1 TO FL*20:NEXT
2550 FOR X=0 TO 15:PALETTE X,B(X
):NEXT X
2560 FOR X=1 TO FL*20:NEXT
2570 AS=INKEYS:IF AS="Q" THEN RE
TURN
2580 GOTO 2530
2590 AS="FLASH RATE":GOSUB320:PR
INT:INPUT" WHAT IS NEW FLASH R
ATE (0-100)";FL:IF FL<0 OR FL>10
0 THEN 2590
2600 GOSUB310:RETURN
2610 'M/L DATA FOR SAVE
2620 DATA142,32,0,23,217,60,189,
225,25,16,142,16,0,166,128,167,1
60,16,140,32,0,37,246,189,224,25
5,16,142,16,0,166,160,173,159,16
0,2,16,140,32,0,37,244,140,160,0
,37,212,57
2630 'DATA FOR LOAD
2640 DATA142,32,0,16,142,16,0,18
9,161,118,167,160,16,140,32,0,37
,245,23,216,45,189,225,25,16,142
,16,0,166,160,167,128,16,140,32,

```

```

0,37,246,189,224,255,140,160,0,3
7,213,57
2650 'LOAD DATA INTO MEMORY
2660 FOR X=&HE00 TO &HE00+47:REA
D A:POKE X,A:NEXT X
2670 FOR X=&HF00 TO &HF00+46:REA
D A:POKE X,A:NEXT X
2680 RETURN
2690 'HOP FACTOR FOR CURSOR
2700 AS="CURSOR MOVEMENT FACTOR"
:GOSUB320
2710 PRINT:PRINT" OLD MOVE FAC
TOR=";JS:PRINT:PRINT" WHAT IS
NEW FACTOR (1-5)"
2720 AS=INKEYS:IF AS="Q" THEN 31
0 ELSE IFAS<"1" ORAS>"5" THEN272
0 ELSE JS=VAL(AS):GOTO 310
2730 KN=0:AS="YOU ARE ABOUT TO E
XIT PROGRAM":GOSUB320:PRINT:PRIN
T" SURE YOU WANT TO DO THAT (Y
/N)?" :POKE65497,0
2740 AS=INKEYS:IFAS="Y" THEN CLS
:POKE65496,0:END ELSE IF AS="N"
THENGOSUB310:GOTO200 ELSE GOTO 2
740
2750 'ERROR ROUTINE
2760 CLOSE:IF ERNO=26 THENAS="FI
LE NAME NOT FOUND ON DISK !":GOS
UB320:GOTO2820
2770 IF ERNO=28 THEN AS="NOT ENO
UGH SPACE LEFT ON DISK !":GOSUB3
20:GOTO2820
2780 IF ERNO=20 THENAS="I/O ERRO
R - CAN'T READ/SAVE FILE":GOSUB3
20:GOTO2820
2790 IF ERNO=1 OR ERNO=4 AND ERL
IN=1480 THENAS="DRAW STRING "+B$
+" DOESN'T MAKE SENSE":GOSUB320:
GOTO2820
2800 IF ERNO=1 THENAS="I CAN'T U
NDERSTAND LINE "+STR$(ERLIN):GOS
UB 320:GOTO2820
2810 AS="****ERROR****":GOSUB320
2820 PRINT:PRINT" ERROR IS NUM
BER ";ERNO;" , IN LINE ";ERLIN
2830 PRINT:PRINT" Press any ke
y to return to program":POKE6549
7,0
2840 IF INKEYS="" THEN2840
2850 GOSUB310:GOTO 200

```

HINT....

Whenever one presses the reset button while running a BASIC program, the screen is cleared and the OK prompt is shown. Wouldn't it be nice to press the reset button and automatically have your program RUN? ie, whenever you press the reset button, the computer will reset the variables and re-RUN your program for you.

Well with the following program, this is possible. Just include it in the first few

Reset Protect

lines of your program followed by the rest of your program.

NOTE: this program will only work on the CoCo 1.

```

10 CLEAR200,31000
20 FOR X=32742 TO 32767:READ
I:POKE X,I:NEXT
30 EXEC32762
40 DATA 58, 142, 58, 18, 16,
50 DATA 222, 33, 48, 140, 246,
60 DATA 159, 166, 28, 175, 127,
70 DATA 255, 64, 126, 173, 192,
80 DATA 48, 140, 236, 159, 114,
90 DATA 57

```

FRICKERS FOLLIES

by Jack Fricker

PUBLIC Domain Software, what it is, what it does and where to get it. Public Domain software is written by people like yourselves who have an interest in OS-9. When these people discovered that they had a need for a particular piece of software and couldn't find just what they wanted by the usual sources, so they sat down and wrote what they needed themselves!

After they had the software written and working to their satisfaction they decided to share their efforts with others. This software is free to all who want to use it. Free to give away, free to use in your own programs, in fact the only thing you are not allowed to do is sell it! This is why it is called Public Domain because it is freely available to the public.

OK. So that answers the first question but leaves the other two, "Where do you get it?" and "What does it do?"

The place to get it is one of the OS-9 user groups. The main one is the Australian OS-9 group. The person to contact for them is Graeme Nichols 02 451 2954 and he will be able to tell you how to join.

You may also be able to find out from the Aust User Group if there is anyone in your area or you may even want to start one of your own.

This is the way that the CoCo user groups started with people meeting and sharing information. This is after all what we are after.

Some of the software is available to those of us with modems. There are some Bulletin Boards around the country that have the programs available for downloading. Some of these may require you to be a member before you can download from them.

Some of these boards may have public domain software that has been downloaded from other Bulletin Boards in other

countries, mainly America.

If you haven't got access to a modem and there is no one near you you can contact me. But please contact the User Group first.

If you want to start one yourselves contact me and I will put your name and address in this column and anyone near you can contact you. But contact the User Group first and let them know as well.

The software that I have comes from the OS-9 User Group in the U.S. and consists of 5 80 track Double sided disks. This amounts to nearly 5 megabytes of software.

The software itself is in various languages, like BASIC09, PASCAL, ASSEMBLER, C, and even COBOL.

All the programs are available in both source and in executable form. This means that you can alter it if you don't like the way it comes. Then re-assemble, Compile or RUN it. Of course if you do make any improvements you can send them in for others to share.

Some of the programs like basic need the RUN-TIME packages for them to work. The RUN-TIME package for BASIC09 is RUNB.

The problem here is that RUNB only comes when you buy BASIC09 which means that you have to buy BASIC09 to use these programs.

Some require even more exotic things to run. For instance if you want to alter the programs written in COBOL you will need the COBOL COMPILER which is very expensive.

Although a CoCo Level II version may be released in the future by TANDY at a more reasonable price which is the way they released C and PASCAL I have not seen or heard of it yet.

However since COBOL is an Assembly Compiler Language the executable version will run ok without any extra purchases. But if you ever wondered what COBOL source code looks like here is your chance.

An Assembly Compiler is one that takes source code like C or

Cobol and then converts or Compiles it into Assembly language source code.

The Assembler source code is then run through a normal assembler which translates it into Machine Code. Machine code just for the record is the actual code that the computer understands and Assembly language is the language that we understand.

I hope that this clears up some of the questions and next month I hope to be able to bring some news about OS9 Level II.

HINT....

Joystick fire button values

If you make programs that require the use of joysticks and, in particular, the fire button, but didn't know what the value of the fire button was, then this is for you.

If both left and right fire buttons are to be kept down, then the value of the fire buttons are either 124 or 252.

eg, P=PEEK(65280):IF P=124 OR P=252 THEN ...'READ BOTH FIRE BUTTONS

If only the left fire button is to be kept down, then the value of the fire button is either 125 or 253.

eg, P=PEEK(65280):IF P=125 OR P=253 THEN ...'READ LEFT FIRE BUTTON

If only the right fire button is to be kept down, then the value of the fire button is either 126 or 254.

eg, P=PEEK(65280):IF P=126 OR P=254 THEN ...'READ RIGHT FIRE BUTTON

If neither of the fire buttons are to be kept down, then the value of the fire buttons are either 127 or 255.

eg, P=PEEK(65280):IF P=127 OR P=255 THEN ...'READ NO FIRE BUTTONS



COLOUR PLUS

GRAPHICS UTILITY

by Brian Ferguson

I HAVE RECENTLY purchased the new CoCo 3. "Fantastic" to say the least would describe my thoughts after a couple of hours at the keyboard. The power of the Palette! (64 colours to choose from). My son calls this command "Palette Power!"

As I previously dabbled with the graphics modes quite a bit on my 'old grey' CoCo, I decided to write a program for the CoCo 3, utilizing as many of the big 'H' commands as possible. The instructions are self-explanatory and it shouldn't be too difficult to memorize the keys and their functions, ie R=Return, T=Text, P=Paint etc.

A few helpful hints!

If your pen becomes hard to locate on the screen, simply press 'I' a few times to enlarge it. Pressing zero reduces pen to one pixel again.

If you're attempting some very intricate 'artwork', press 'C' to obtain non auto-repeat movement.

The 'Q' key option: There are 2 selectable speeds (the 'S' key) which vary the blink rate, but if you wish to vary this mode, edit line 10: BLINK=5. The higher the number, the slower the rate.

Have fun!

Art Section:

Key:	Description:
Arrow Keys	:move Pen
A,Z,X,D ..	:move Pen
Arrow Keys (shifted)	:move pen 20 pixels
1-8	:selects ink colour
1-8 (shifted)	:Selects ink colour
0	:reduces pen to one pixel size
=	:clears screen to pen (ink) colour.
E	:erases info section
X	:prints info section
M	:marks a point
R	:returns pen to marked point
T	:selects text option
Y	:prints pen location
I	:increases pen size
O	:decreases pen size

P	:paints
S	:hi/lo speed
B	:draws a block from marked point to pen
F	:fills in block
L	:draws line from pen to marked point
'.'	:sets a block (pen size)
F1	:can be used to change ink colour
Q	:turns ink colour off-on @ approx. 1/2 sec. intervals. (press CTRL to exit)
Clear	:lifts pen
Enter	:lowers pen
ALT	:alter palette colour
CTRL	:returns palette to original form

Circles:

Using the joystick: Press 'J' then press fire button. Hold fire button down to move pen. Release button to draw circles.
 Press 'I' to increase circle.
 Press 'O' to decrease circle.
 Press F1 to change ink colour.
 Pen up:Top half of circle drawn.
 Pen down:Full circles drawn.

Using the 'U' key:Mark point, move pen vertically from marked point then press 'U' key.
 Pen up:left side of circle drawn.
 Pen down:right side is drawn.
 OR
 Move pen horizontally from mark, point, then pres 'U' key.
 Pen up:top half of circle drawn.
 Pen down:lower half drawn.

GETting and PUTting

Mark point at top or bottom on the left side of rectangle to get. Move pen to diagonal corner and press 'G'. Move pen to new location then press 'H' to put it. Arrow keys will move area about screen. To move area diagonally, press two arrow keys.
 (Press CTRL-O to exit)

Auto Key

Press 'C' key to select between auto repeat and non auto-repeat. 'a/k' is displayed if auto key is on.

Text Section

(Obtained by pressing 'T' key)

Key:	Description:
F1	:selects foreground (ink) colour.
F2	:selects background (ink) color.
Arrow Keys	:move cursor to new location.
Clear	:press clear then use arrow keys to move cursor one pixel.
CTRL	:exits text
Up arrow (shifted)	:enlarge print size.
Down Arrow (shifted)	:decrease print size.
Clear(shifted)	:PRETTY PRINT MODE (press shift clear again to exit this mode).

There are 4 print sizes to pick from. If print and border colours are identical then printing is superimposed. Also, there is no lowercase.

Break Key: Ends program.

The Listing:

```

1 ' *COLOUR PLUS*
   BRIAN FERGUSON
   JAN/87
2 GOTO10
3 SAVE"157:3":END
4 SAVE"COLOUR+":END
5 '
6 '
7 ' GRAFIX PROGRAM FOR COCO 3
8 '
9 '
10 BLINK=5
20 ONBRKGOTO1360
30 WIDTH40:CLS2:ATTR2,1:LOCATE13
,9:PRINT"*COLOUR PLUS*";:LOCATE1
8,12:PRINT"By";:LOCATE13,15:PRIN
    
```

T"Brian Ferguson":LOCATE19,17:FO
RD=1TO1000:NEXT:GOTO930
40 JQ=1
50 IFJ+I>TF THENJ=TF-I ELSEIFJ<0
THENJ=0
60 IFL+I>TB THENL=TB-I ELSEIFL<0
THENL=0
70 IFJQ=1THENG=1-GG:HGET(J,L)-(
J+I,L+1),1:HCOLORG:HLINE(J,L)-(
J+I,L+1),PSET,BF:HPUT(J,L)-(J+I,
L+1),1,PSET:RETURN
80 HCOLORG:HLINE(J,L)-(J+I,L+1),
PSET,BF:RETURN
90 A\$=INKEY\$:IFAS=""THEN620ELSEA
S=ASC(A\$):IFAS=9GOSUB50:J=J+1:GO
SUB50:GOTO90ELSEIFAS=8GOSUB50:J=
J-1:GOSUB50:GOTO90ELSEIFAS=10GOS
UB50:L=L+1:GOSUB50:GOTO90ELSEIFA
S=94GOSUB50:L=L-1:GOSUB50:GOTO90
100 IFAS=65THENJ=J+1:L=L+1:GOSUB
50:GOTO90ELSEIFAS=90THENJ=J-1:L=
L-1:GOSUB50:GOTO90ELSEIFAS=68THE
NJ=J-1:L=L+1:GOSUB50:GOTO90ELSEI
FAS=88THENJ=J+1:L=L-1:GOSUB50:GO
TO90
110 IFAS=13THENJQ=0:GOSUB1090:GO
TO90ELSEIFAS=12THEN380ELSEIFAS=6
7THENFC=0:HCOLOR4:HPRINT(7,24),"
a/k":GOTO130
120 GOTO140
130 IFFC=1THEN90ELSEA\$=INKEY\$:IF
A\$=""THEN620ELSEAS=ASC(A\$)
140 IFAS=13THENJQ=0:GOSUB1090:GO
TO130
150 IFAS=95THENL=L-20:GOSUB50:GO
TO130ELSEIFAS=91THENL=L+20:GOSUB
50:GOTO130ELSEIFAS=21THENJ=J-20:
GOSUB50:GOTO130ELSEIFAS=93THENJ=
J+20:GOSUB50:GOTO130
160 IFPEEK(344)=247GOSUB50:J=J+1
:GOSUB50:GOTO160
170 IFPEEK(343)=247GOSUB50:J=J-1
:GOSUB50:GOTO170
180 IFPEEK(342)=247GOSUB50:L=L+1
:GOSUB50:GOTO180
190 IFPEEK(341)=247GOSUB50:L=L-1
:GOSUB50:GOTO190
200 IFPEEK(339)=254GOSUB50:J=J+1
:L=L+1:GOSUB50:GOTO200
210 IFPEEK(342)=254GOSUB50:J=J-1
:L=L+1:GOSUB50:GOTO210
220 IFPEEK(338)=247GOSUB50:J=J+1
:L=L-1:GOSUB50:GOTO220
230 IFPEEK(340)=247GOSUB50:J=J-1
:L=L-1:GOSUB50:GOTO230
240 IFAS=67THENFC=1:HCOLOR8:HLIN
E(52,184)-(86,191),PSET,BF:GOTO9
0ELSEIFAS=71THEN550
250 IFAS=72THENPOKE65497,0:GOTO6
60ELSEIFAS=48THENI=0:U=0
260 IFAS=84THENNI=I:CF=FC:FC=1:G
E=G:POKE65497,0:GOTO920
270 IFAS=89GOSUB1080ELSEIFAS=76T
HEN HCOLORG:HLINE(XR,YR)-(J,L),P
SET
280 IFAS=81THEN1330ELSEIFAS=103G
OSUB1010:GOTO130
290 IFAS=64THEN980:ELSEIFAS=46AN
DJQ=0THEN HCOLORG:HLINE(J,L)-(J+
I,L+1),PSET,BF:GOTO130
300 IFAS=69THENHCOLORG:HLINE(0,
184)-(319,191),PSET,BF:ELSEIFAS=
189THEN PALETTEGB:JK=0:GOSUB126
0:GOSUB1350
310 IFAS=83THENP=1-P:POKE65496+P
,0:SOUND1,P+1:GOTO130
320 IFAS=73ANDI<32THENI=I+1:GOTO

130
330 IFAS=79ANDI>0THENI=I-1:GOTO1
30
340 IFAS=77THEN XR=J:YR=L:GOSUB1
080:GOTO130ELSEIFAS=85GOSUB1270
350 IFAS=32ANDAS<41THENG=AS-25:G
OSUB1020:GOTO130ELSEIFAS=48ANDAS
<57THENG=AS-49:GOSUB1020:GOTO130
360 IFAS=74THENI1=I:0:GOTO410
370 IFAS=70THEN HCOLORG:HLINE(XR
,YR)-(J,L),PSET,BF ELSEIFAS=66TH
EN HCOLORG:HLINE(XR,YR)-(J,L),PS
ET,B
380 IFAS=12THEN HCOLOR4:HPRINT(1
9,24),"PENUP":GOSUB40:GOTO130ELS
EIFAS=61GOSUB530
390 IFAS=75GOSUB1050:GOSUB1020:E
LSEIFAS=80THEN HPAINT(J+I+1,L),G
,G:GOSUB1050:GOSUB1020: ELSEIFAS
=82THEN J=XR:L=YR
400 GOTO130
410 GD=BUTTON(0):IFGD=1THEN420EL
SE410
420 J=JOYSTK(0):J=J*5.05
430 L=JOYSTK(1):L=L*2.92
440 IFJQ=1THENE1=.50ELSEE1=0
450 GD=BUTTON(0):IFGD=1GOSUB630:
GOSUB650:HCIRCLE(J,L),0,G:GOSUB6
40:GOTO420
460 HCIRCLE(J,L),U,G,1,E1,0
470 K\$=INKEY\$:IFK\$=""THEN420
480 IFPEEK(339)=253THENU=U+1:GOT
0520
490 IFPEEK(345)=253ANDU>0THENU=U
-1:GOTO520
500 IFK\$=CHR\$(12)THENJQ=1:GOSUB1
040:GOTO470ELSEIFK\$=CHR\$(13)THEN
JQ=0:GOSUB1090:GOTO470ELSEIFK\$=C
HR\$(103)GOSUB1010:GOTO470ELSEIFK
\$=""0"THENU=0:GOTO470
510 IFK\$<>"I"ANDK\$<>"O"THENJ=INT
(J):L=INT(L):I=11:AS=ASC(K\$):GOT
0140ELSE420
520 HCIRCLE(J,L),U,G,1,E1,0:GOTO
480
530 HCLSG:GY=G:CH=E(G)
540 GOSUB1050:GOSUB1020:RETURN
550 XR=INT(XR/2)*2:IFJ>XR THENMM
=INT((J-XR)/2)*2:ELSE130
560 IFL>YR THENNN=L-YR:ELSENN=YR
-L
570 IFL=YR THEN130
580 IFMM*NN)14050THENHLINE(192,1
84)-(319,191),PSET,BF:FORD=1TO15
:HCOLORPS:HPRINT(24,24),"PLEASEr
educeSIZE":FORVV=1TO200:NEXT:PS=
1-PS:NEXT:GOSUB1050:GOSUB1020:GO
TO130
590 AQ=0:SOUND200,1:HGET(XR,YR)-
(J,L),2:GOTO130
600 IFAQ=1THEN130
610 HPUT(J,L)-(J+MM,L+NN),2,PSET
:RETURN
620 POKE65496,0:GOSUB630:BZ=1-BZ
:GOSUB650:GOSUB640:POKE65496+P,0
:IFFC=1THEN90ELSEGOTO130
630 HGET(J,L)-(J+(I+2),L+(I+2)),
1:RETURN
640 HPUT(J,L)-(J+(I+2),L+(I+2)),
1,PSET:RETURN
650 HCOLORBZ,BZ:HLINE(J,L)-(J+I,
L+1),PSET,BF:RETURN
660 IFAQ=1THEN130
670 IFPEEK(135)=189THENPOKE135,0
:POKE65496+P,0:GOTO130
680 IFPEEK(341)=247THENL=L-2
690 IFPEEK(342)=247THENL=L+2

700 IFPEEK(343)=247THENJ=J-2
710 IFPEEK(344)=247THENJ=J+2
720 IFJ+MM)319THENJ=319-MM ELSEI
FJ<0THENJ=0
730 IFL+NN)181THENL=183-NN ELSEI
FL<0THENL=0
740 GOSUB600:GOTO670
750 IFCE<G THENHCOLORCB:HLINE(J
,L)-(J+IX+WD,L+1),PSET,BF
760 FORD=0TOWD:HDRAWPS+"BM"+STR\$(
J+2)+","+STR\$(L+I-2)+PP\$+BB\$(KS
-32):J=J+1:NEXT:J=J+IX
770 IFJ+IX+WD-1>TF THENL=L+I+1:J
=0
780 IFL+I>TB THENL=TB-I
790 K\$=INKEY\$:IFK\$=""THEN900ELSE
KS=ASC(K\$)
800 IFKS)31ANDKS<91THEN750ELSEIF
KS=92THENPP=1-PP
810 IFKS=13THENJ=0:L=L+I+1:GOTO7
80
820 IFKS=4THEN910ELSEIFKS=103GOS
UB1010:GE=G:ELSEIFKS=95ANDS<16TH
ENS=S+4:L=L-6:ELSEIFKS=91ANDS>4T
HENS=S-4:L=L+6:ELSEIFKS=189THENI
=NI:J=INT(J):L=INT(L):FC=CF:POKE
65496+P,0:GOSUB1100:GOSUB1030:GO
TO130
830 IFKS=8THENJ=J-(1+IX+WD)ELSEI
FKS=9THENJ=J+1+IX+WD ELSEIFKS=94
THENL=L-(I+1):ELSEIFKS=10THENL=L
+I+1
840 IFKS=12THEN950
850 IFS<12THENIX=S+3ELSEIX=S+4
860 WD=S/4*PP:I=S/4*6+4:P\$="S"+S
TR\$(J):PP\$="C"+STR\$(GE)
870 IFL<1THENL=0ELSEIFL+I>TB THE
NL=TB-I
880 IFJ<1THENJ=0ELSEIFJ+IX+WD>TF
THENJ=TF-(S+IX+WD)
890 IFBX=1THENRETURNELSE790
900 GOSUB630:BZ=1-BZ:HCOLORBZ:HL
INE(J,L)-(J+IX+WD,L+1),PSET,BF:G
OSUB640:IFL9=1THEN950ELSE790
910 CE=CE+1:IFCE>15THENCE=0
920 HCOLOR8:HLINE(51,184)-(70,19
1),PSET,BF:HCOLOR4:HPRINT(6,24),
STR\$(CE+1):HCOLORCB:HLINE(71,184
)-(81,191),PSET,BF:GOTO850
930 POKE65497,0:HBUFF1,700:HBUFF
2,7200:DIND(16),E(16),BB\$(58):FO
RXT=0TO58:READBB\$(XT):NEXT:P=1:G
=0:TB=183:A=22:D5=8:S=4:TF=319:A
Q=1:J=159:L=88:JQ=1:JK=1:GY=8:Z=
1:GOSUB1350
940 IX=7:FC=0:AC=1:PALETTEGB:HS
GREEN2:HCLSO:GOSUB1050:GOSUB1110
:GOSUB1050:CH=0:GOTO130
950 BX=1:L9=1:K\$=INKEY\$:IFK\$=""T
HEN900ELSEKS=ASC(K\$)
960 IFKS=94THENL=L-1ELSEIFKS=10T
HENL=L+1ELSEIFKS=8THENJ=J-1ELSEI
FKS=9THENJ=J+1
970 GOSUB870:IFKS<8ORKS)10ANDKS<
>94THENL9=0:BX=0:GOTO800ELSE950
980 IFPEEK(135)=189THENPOKE135,0
:CH=E(GY):GOTO130
990 RN=RND(64)-1:PALETTEPR,RN:E(C
PR)=RN:PR=PR+1:IFPR=GY THENPR=GY
+1
992 IFPR>15THENPR=0
1000 GOTO980
1010 G=G+1:IFG>15THENG=0
1020 HCOLOR8:HLINE(0,184)-(39,19
1),PSET,BF:HCOLOR4:HPRINT(2,24),
STR\$(G+1):HPRINT(1,24),"C-":HCOL
ORG:HLINE(40,184)-(50,191),PSET,


```

BF
1030 IFFC=0THENHCOLOR4:HPRINT(7,
24),"a/k"
1040 IFJQ=1THENHCOLOR4:HPRINT(19
,24),"PENUP":RETURNELSERETURN
1050 HCOLOR8:HLINE(0,184)-(319,1
91),PSET,BF
1060 MR=0:FORD=0TOD5:HCOLOR:HLINE(193+MR,184)-(201+MR,191),PSET
,BF:MR=MR+8:NEXT:D5=15:IFHY=0THE
NRETURN
1070 GOSUB1020:IFHY=1GOSUB1260
1080 HCOLOR8:HLINE(80,184)-(151,
191),PSET,BF:HCOLOR4:FG$=RIGHT$(
STR$(L),3):HPRINT(10,24),STR$(J)
+","FG$:RETURN
1090 HCOLOR8:HLINE(152,184)-(192
,191),PSET,BF:RETURN
1100 HCOLOR8:HLINE(52,184)-(81,1
91),PSET,BF:RETURN
1110 D=9:HY=1:HCOLOR2:HPRINT(13,
2),"*COLOUR PLUS*":HCOLOR4:HPRINT
T(10,4),"PALETTE = 16 COLOURS,"
:HPRINT(4,6),"9 OF WHICH HAVE BEE
N PRESELECTED.":HPRINT(3,9),"CHO
OSE REMAINING 7 FROM 63 COLOURS"
1120 HPRINT(3,15),"UP,DOWN ARRO
W KEYS SCROLL COLOURS":HPRINT(4,
17),"LEFT,RIGHT ARROW KEYS MOVE
CURSOR"
1130 HCOLOR1:HPRINT(4,20),"SPACE
BAR ADDS CHOICE TO PALETTE":HCO
LOR3:HPRINT(4,21),"-----"
1140 GOSUB1250:KK=1:MR=0:HCOLOR8
:HPRINT(3,13),"":HGET(A,103)-(A
+10,113),1
1150 AS=INKEY$:IFAS=""THEN1150EL
SEHCOLOR0:HLINE(A,103)-(A+10,113
),PSET,BF
1160 IFAS=CHR$(10)GOSUB1250
1170 IFAS=CHR$(94)THENZ=Z-14:GOS
UB1240

```

```

1180 IFAS=CHR$(8)THENA=A-45:KK=K
K-1:ELSEIFAS=CHR$(9)THENA=A+45:K
K=KK+1
1190 IFA<22THENA=22ELSEIFA>290TH
ENA=290
1200 HPUT(A,103)-(A+10,113),1,PS
ET
1210 IFAS=CHR$(32)GOSUB1230:IFD>
15THENHCLS8:JK=0:GOSUB1260:RETUR
N
1220 GOTO1150
1230 HCOLOR1:HLINE(158,70)-(168,
80),PSET,BF:HCOLOR3:HPRINT(19,9)
,15-D:D(D)=KK:E(D)=KK:D=D+1:RETU
RN
1240 IFZ<1THENZ=8:RETURNELSEY=2:
FORX=9TO15:PALETTE,X,Z:HCOLORX:HL
INE(Y,86)-(Y+45,96),PSET,BF:Y=Y+
45:Z=Z+1:NEXT:KK=KK-7:RETURN
1250 IFZ>57THENRETURNELSEY=2:FOR
X=9TO15:PALETTE,X,Z:HCOLORX:HLIN
E(Y,86)-(Y+45,96),PSET,BF:Y=Y+45:
Z=Z+1:NEXT:KK=KK+7:RETURN
1260 IFJK=1THENRETURNELSEFORD=9T
O15:PALETTE,D(D):E(D)=D(D):NEXT
:JK=1:RETURN
1270 IFJ>XR THENXD=(J-XR)/2:XM=X
D+XR:GOTO1310
1280 IFJ<XR THENXD=(XR-J)/2:XM=X
D-J:GOTO1310
1290 IFL>YR THENYD=(L-YR)/2:YM=Y
D+YR:GOTO1320
1300 IFL<YR THENYD=(YR-L)/2:YM=Y
D-L:GOTO1320
1310 IFJQ=1THENHCIRCLE(XM,L),XD,
G,1,.5,0:RETURNELSEHCIRCLE(XM,L)
,XD,G,1,0,.5:RETURN
1320 IFJQ=1THENHCIRCLE(J,YM),YD,
G,1,.25,.75:RETURNELSEHCIRCLE(J,
YM),YD,G,1,.75,.25:RETURN
1330 HC=E(G)
1340 PALETTEG,CH:FORD=1TO46*BL:N

```

```

EXT:PALETTEG,HC:FORD=1TO46*BL:NE
XT:IFPEEK(135)=189THENPOKE135,0:
GOTO130ELSE1340
1350 E(0)=18:E(1)=54:E(2)=9:E(3)
=36:E(4)=63:E(5)=27:E(6)=45:E(7)
=38:E(8)=0:CH=E(GY):RETURN
1360 POKE65496,0:WIDTH32:PALETTE
RGB:END
1370 DATA BR4,BR2U0BU2U4,BRBU6D2
BR2U2BEBD6,BRU2LR4LD2BL3BU4RU2D2
R2U2D2RBD4,BUR2DUREHL2HERUDR2BD5
,BUE4BL3LURDBR3BD5URDL,BR4BU2G2L
HE3UHLGDF4,BR2BU6D2BR2BD4,BR4BU6
LGD4FR,REU4HLBR4BD5,BUE4G2U2D4U2
L2R4L2H2F4BD,BU3R4L2U2D4BR2BD
1380 DATABR3BULURD2GRR2BU,BRBU3R
2BRBD3,BR2LURDBR2,UE4UBD6,BUU4ER
2FD4GL2HBR4BD,BRU5ED6LR2BR,BU5E
R2FDG4R4,BU5ER2FDGLRFDGL2HBR4BD,
BR3UGG3R4BD3,BUFR2EUHL3U3R4BD6,B
U3R3FDGL2HU3E2RBRBD6,BU6R4G3D3BR
3,BR4BU2DGL2HUER2L2HUER2FDGFB2
1390 DATABUFR2EU4HL2GDFR2BRBD3,B
RBU5LURDBR2BD4,BR4BU6G3F3,BRU4
R2BL2BD2R2BR3BD2,E3H3BR4BD6,BU5E
R2FDG2BD2UBR2BD,BU5ER2FD4GL2HUER
3BD3,U4E2F2D2L4R4D2,RU6LR3FDGL2R
2FDGL3BR4,BR4BUGL2HU4ER2FBD4
1400 DATARU6LR3FD4GL2BR3,U3R4L4U
3R4BD6L4R4,U3R4L4U3R4BD6,BR2BU3R
2D2GL2HU4ER2FBD5,U6D3R4U3D6,BRR2
LU6LR2BRBD6,BU2DFR2EU5BD6,U6BR4G
3F3,R4L4U6BR4BD6,U6F2E2D6,U6DF4U
5D6,R4L4U6R4D6,U6R3FDGL3BR4BD3,B
UU4ER2FD4GL2HBR2B2UF2
1410 DATAU6R3FDGL3RF3,BUFR2EUHL2
HUER2FBD5,BU6R4L2D6BR2,U6D6R4U6D
6,BU6D4F2E2U4BD6,U6D6E2F2U6D6,UE
2H2UDF2E2UDG2F2D,BU6DF2E2UDG2D3B
R2,BU6R4DG4DR4

```

COCO 3 UTILITIES

continued from page 16

```

contains ";BY;" bytes":PRINT"Pr
ess <SPACEBAR> to pause, <D> for
directory, <R> to restart":PRIN
TF1$1110 FORX=1TOLOF(1):GET#1,X
1120 I$=INKEY$:IFIS<>""THEN1170
1130 PRINT#DE,H$;
1140 NEXT
1150 PRINTF1$:PRINTTAB(20);:LINE
INPUT** Press <ENTER> to displa
y directorSSZ-j1060 CLOSE
1160 GOSUB690:GOTO510
1170 IFIS<>"" THEN1190
1180 IFINKEY$=""THEN1180ELSE1130
1190 IFIS="R"THEN1100
1200 IFIS="D"THEN1230ELSE1130
1210 '
1220 IFINSTR(P,FIS,"/")=0THENFL=
1:GOTO1270
1230 B=INSTR(P,FIS,"/")
1240 FI=VAL(MID$(FIS,P,B-P))
1250 P=B+1:IFB=LEN(FI$)THENFL=1
1260 GOSUB720
1270 RETURN
1280 PRINT:PRINT"Use slashes to
separate more than one file... 1
5/3/21/10/ etc."
1290 PRINT"Ensure that a slash '

```

```

/' is placed after the last file
No. ie. 10/"
1300 RETURN
1310 '
1320 'SCREEN DRIVER ROUTINES FOR
CM-8 MONITOR
1330 '
1340 'BLUE TEXT ON BLACK BACKGRO
UNG
1350 PALETTE13,0:PALETTE12,48:PA
LETTE0,0:PALETTE14,24
1360 CLS1:CLS0:ATTR6,0:WIDTH80
1370 PALETTE8,0:PALETTE9,48:PALE
TTE10,0:PALETTE11,48
1380 '
1390 'GREEN TEXT ON BLACK
1400 'PALETTE13,0:PALETTE12,48:P
ALETTE0,0:PALETTE14,22
1410 'CLS1:CLS0:ATTR6,0:WIDTH80
1420 'PALETTE8,0:PALETTE9,48:PAL
ETTE10,0:PALETTE11,48
1430 '
1440 'SCREEN DRIVER FOR GREEN SC
REEN MONITOR
1450 'PALETTE0,0:PALETTE1,16:PAL
ETTE2,32:PALETTE3,48
1460 'PALETTE4,32:PALETTE5,16:PA
LETTE6,0:PALETTE13,63
1470 'PALETTE0,0:WIDTH80:CLS1:AT
TR3,0
1480 RETURN
1490 '
1500 'ERROR TRAPPING ROUTINES
1510 IF ERNO<>4 THEN 1520 ELSE 1

```

```

620
1520 IF ERNO<>26 THEN 1530 ELSE
1680
1530 IF ERNO <>20 THEN 1540 ELSE
1650
1540 IF ERNO <>33 THEN 1550 ELSE
1680
1550 CLS1:CLS0:ATTR6,0
1560 '
1570 'ERROR MESSAGES
1580 PRINT"FATAL ERROR HAS OCCUR
ED":PRINT
1590 PRINT"ENSURE THAT THE TASK
THAT YOU HAVE JUST ATTEMPTED"
1600 PRINT"HAS IN FACT BEEN COMP
LETED CORRECTLY"
1610 GOTO 1710
1620 PRINT"NO FILE ON DISK":PRIN
T
1630 PRINT"PLEASE WAIT... THEN S
ELECT ANOTHER!"
1640 GOTO 1710
1650 CLS1:CLS0:ATTR6,0
1660 PRINT"DISK I/O ERROR... IS
DISK FORMATTED?"
1670 GOTO1710
1680 CLS1:CLS0:ATTR6,0
1690 PRINT"FILE ALREADY EXISTS.
. NO COPY MADE"
1700 GOTO 1630
1710 FOR ZP=1 TO 5000:NEXT
1720 GOTO 270
1730 'END OF PROGRAM

```

PAST and PRESENT COCO



by Leigh Dawes

THE IDEA for making this program was to demonstrate the differences between the CoCo 2 and the CoCo 3. Just load and run the program following the prompts. This program is to enter the COCO3 competition.

The Listing:

```
1 ON BRK GOTO2390
2 GOTO50
3 SAVE"142:3":END
4 IS=INKEYS:IFIS=""THEN5ELSERETU
RN
5 FORDE=1TO1000:NEXT:RETURN
6 PRINT"PRESS ANY KEY";:RETURN
7 FORL=1TO22:READX,Y,AS:IFX=-1TH
ENL=22ELSELOCATEX,Y:PRINTAS
8 NEXT:RETURN
9 HPRINT(X,Y),"Press any key to
continue":RETURN
10 HLINE(0,155)-(X,Y),PRESET,BF:
RETURN
11 A=X+5:B=Y+5:HPAINT(A,B),C,C:R
ETURN
12 X$="BM"+STR$(X)+","+STR$(Y):C
=(C+1)AND&HOF:HCOLORC:RETURN
13 **PROGRAMMED BY LEIGH DAVES
14 **PROGRAM NAME:
15 *****PAST TO PRESE
NT
16 **FOR COCO3 COMPETITION
17 **ON THE COCO3 128K
18 **14/11/86
19 **USED RSDOS1.0
20 **TV PHILLIPS TOWN & COUNTRY
34
21 **TW64 NAMED "TEXT" IN ASCII
&BIN
50 PALETTECMP:WIDTH32:FORX=1TO15
:READAS:IFAS=""**THENX=15ELSEIFA
S=""**THENGOSUB6:GOSUB6:GOSUB6:GO
SUB6ELSEPRINTAS
70 NEXT
80 DATA " ", " ", "This is the norma
l coco2,screen which has 32 char
acters,across and 16 lines down,
The new coco3 has 40 and 80, cha
racter screens,*, " ", "Now we have
the PALETTE command,**
90 PALETTE12,60:PALETTE13,10:GOS
UB6:GOSUB6
100 PRINT:PRINT"And it does have
lowercase":POKE&HFF22,PEEK(&HFF
22)OR16:GOSUB6:GOSUB6:GOSUB6:PRI
NT"Or inverse video":POKE&HFF22,
PEEK(&HFF22)OR48:GOSUB6:GOSUB6:P
RINT@424,,:GOSUB7:GOSUB5
```

```
110 WIDTH40:GOSUB8
120 DATA0,4,This is the WIDTH40
screen and you will,0,8,realise
that it supports 40 characters,0
,12,per line. It also supports
Attributes,0,16,for each charact
er printed.,-1,-1,A
130 LOCATE10,22:ATTR2,1,B:GOSUB7
:ATTR6,0:GOSUB5
140 WIDTH80:GOSUB8
150 DATA0,4,This is the WIDTH80
screen. You will notice that th
e display is not too clear,0,8,u
nless you have a good quality TV
or an RGB Analog Monitor. This
mode also,0,12,supports the ATT
RIBUTE command.,-1,-1,A
160 LOCATE30,22:ATTR2,1,B:GOSUB7
:ATTR6,0:GOSUB5
170 PALETTE13,18:PALETTE12,0:WID
TH32:PRINT@33,"THE FOLLOWING IS
THE HIGHEST RESOLUTION AVAILA
BLE IN COCO2'S PMODE SCREENS."
180 POKE65497,0:PMODE4,1:PCLS:F
ORR=10TO90STEP10:CIRCLE(128,96),
R:NEXT:PRINT@423,,:GOSUB7:GOSUB5
:SCREEN1,0:POKE65496,0:GOSUB6:GO
SUB6190 CLS:PRINT@224,"WOULD YOU
LIKE TO SEE THAT AGAIN(Y)ES OR
(N)O?":GOSUB5:IFIS="Y"THEN170
191 CLS:PRINT@224,"HERE IT IS AG
AIN IN ALL COLORS AVAILABLE":GO
SUB6:GOSUB6:SCREEN1,0:FORB=0TO63
:PALETTE8,B:IFB<30THENPALETTE9,6
3ELSEPALETTE9,0
192 FORDE=1TO80:NEXT:NEXT:PALETT
E8,0:PALETTE9,18:GOSUB6:GOSUB6
200 WIDTH40:LOCATE0,10:PRINT"The
following is the 320x192 resolu
tion graphics screen. Please no
te that thereare 16 colors avail
able out of a 64 color palet
te."
210 LOCATE10,22:ATTR2,1,B:GOSUB7
:ATTR6,0:GOSUB5
220 HSCREEN2:POKE65497,0:FORS=0T
O15:PALETTES,S*4:NEXT:FORC=0TO15
:HCOLORC:HIRCLE(C*10+50,96),50:
HPAINT(C*10+50,96),C,C:NEXT:POKE
65496,0
230 HPRINT(1,20),"Also notice th
at it prints on the":HPRINT(1,21
),"graphics screen."
240 HCOLOR12:HPRINT(16,21)," in"
:HCOLOR4:HPRINT(19,21)," any":HC
OLOR8:HPRINT(24,21),"4. color."
250 HCOLOR10:X=5:Y=22:GOSUB10:GO
SUB5:X=319:Y=191:GOSUB11
250 HPRINT(2,20),"By using the P
ALETTE command you":HPRINT(0,21
),"could do this.":HPRINT(10,22),
```

```
"PALETTE slot,color"
280 FORC=1TO20:FORS=1TO15:PALETT
ES,RND(63):NEXT:FORDE=1TO300:NEX
T:NEXT:X=319:Y=191:GOSUB11
290 X=5:Y=22:GOSUB10:GOSUB5
300 PALETTECMP:X=319:Y=191:GOSUB
11
320 HPRINT(2,20),"By using the P
ALETTE command you can":HPRINT(0
,21),"animate":X=5:Y=22:GOSUB10:
GOSUB5:X=319:Y=191:GOSUB11
330 FORS=0TO15:PALETTES,0:NEXT
340 FORC=1TO5:FORS=1TO15:PALETTE
S,63:FORDE=1TO100:NEXT:PALETTES-
1,0:NEXT:FORDE=1TO400:NEXT:PALET
TES-1,0:NEXT
345 PALETTECMP
350 HPRINT(0,20),"How's that for
another form of animation"
355 HPRINT(0,21),"That was a hal
fmoon into fullmoon??"
360 X=5:Y=22:GOSUB10:GOSUB5:PALE
TTECMP
370 WIDTH40:PALETTE0,0:HSCREEN2:
HCOLOR8:HDRAW"BM160,20":HLINE-(1
30,120),PSET:HLINE-(190,40),PSET
:HLINE-(130,40),PSET:HLINE-(190,
120),PSET:HLINE-(160,20),PSET
380 HPAINT(160,30),1,8:HPAINT(13
5,45),2,8:HPAINT(135,110),3,8:HP
AINT(185,110),4,8:HPAINT(185,45)
,5,8:HPAINT(160,70),6,8
385 HCOLOR7:HPRINT(5,20),"Twinkl
e, twinkle, little star."
390 FORC=1TO100:FORS=1TO6:PALETT
ES,RND(64)-1:NEXT:NEXT
400 FORS=0TO15:PALETTES,0:NEXT
405 X=319:Y=191:GOSUB11:PALETTE7
,60:HPRINT(2,20),"More animation
."
410 FORC=1TO20:FORS=1TO5:PALETTE
S,60:PALETTES-1,0:FORDE=1TO50:NE
XT:NEXT:PALETTES-1,0:NEXT
420 X=319:Y=191:GOSUB11:HPRINT(2
,20),"How about some colour"
430 FORC=1TO15:FORS=1TO5:PALETTE
S,RND(63):PALETTES-1,0:FORDE=1TO
50:NEXT:NEXT:PALETTES-1,0:NEXT
500 HSCREEN2:FORS=0TO15:PALETTES
,S*4:NEXT:C=1
505 X=0:Y=0
507 READAS:IFAS=""-1"THEN600
508 C=C+1:IFC>15THENC=1
509 HCOLORC
510 HPRINT(X,Y),AS:X=X+1:IFX>39T
HENX=0:Y=Y+1:IFY>22THEN600
520 DATAA,s," ",y,o,u," ",c,a,n,
" ",s,e,e," ",t,h,i,s," ",i,s,"
",a," ",v,e,r,y," ",v,e,r,s,a,t,
i,l,e," ",c,o,m,p,u,t,e,r,,-1,*
522 DATAW,i,t,h," ",1,2,8,K," ",
R,A,M," ",a,n,d," ",2,1," ",n,e,
```

```

w," ",c,o,m,m,a,n,d,s,..-1,*
524 DATAE,x,t,r,a," ",k,e,y,s,"
",b,y," ",d,e,m,a,n,d,..-1,*
526 DATAA,n,d," ",l,o,o,k," ",a,
t," ",t,h,e,s,e," ",c,o,l,o,r,s,
..-1,*
528 DATAT,h,e,y," ",c,h,a,n,g,e,
" ",w,i,t,h,o,u,t," ",a," ",f,l,
i,c,k,e,r,..-1,*
529 DATAT,r,y," ",i,t,..",E,n,
t,e,r," ",y,o,u,r," ",o,w,n," ",
m,e,s,s,a,g,e,..-1,-2
530 GOTO507
600 FORC=1TO10:FORS=1TO15:PALETT
ES,RND(63):NEXT:NEXT
610 READA$:IFAS<>"-2"THENX=0:Y=Y
+2:GOTO507
615 IFFL=1THEN700ELSEFL=1
620 C=1:X=5:Y=15
630 GOSUB5:IFIS=CHRS(13)ORX)39TH
EN700ELSEHPRINT(X,Y),IS
640 C=C+1:IFC>15THENC=1
645 HCOLORC
650 X=X+1
660 GOTO630
700 POKE65497,0:PALETTECMP:WIDTH
40:HSCREEN4:HCOLOR2:FORX=1TO100:
HLINE-(RND(640)-1,RND(150)),PSET
:NEXT
710 HPRINT(5,20),"How's that for
resolution and speed."
720 HPRINT(5,21),"This is the 64
0x192 Hires graphics screen."
730 HCOLOR1:X=30:Y=22:GOSUB10
740 GOSUB5
750 HCOLOR2,0:HCLS
760 FORP=1TO90:FORS=1TO3:HCOLORC
:HLINE(P*30-30+C*10,C*10)-(P*30-
30+C*10+20,C*10+20),PSET,BF:NEXT
:NEXT
770 C=1:HCOLORC:FORX=10TO80STEP1
0:HCIRCLE(319,96),X:NEXT:FORX=10
TO80STEP10:HPAINT(319-X+5,96),C,
1:C=C+1:IFC>3THENC=1
780 NEXT
790 FORP=1TO30:FORS=1TO3:FORS=1T
O3:PALETTES,S*C*7 AND3:NEXT:FORD
E=1TO100:NEXT:NEXT:NEXT
800 HPRINT(5,22),"One can add an
y color required from 64 colours
"
805 PALETTE0,0
810 FORP=1TO90:FORS=1TO3:PALETTE
S,RND(64)-1:FORDE=1TO50:NEXT:NEX
T:NEXT
820 X=639:Y=191:GOSUB11
900 X=30:Y=22:GOSUB10:GOSUB5
1000 C=0:POKE65497,0:HSCREEN2:FO
RS=0TO15:PALETTES,S*4:NEXT
1010 X=10:Y=10:GOSUB1170
1020 X=X+50:GOSUB1180
1030 X=X+50:GOSUB1190
1040 X=X+50:GOSUB1200
1050 X=X+50:GOSUB1210
1060 X=10:Y=100:GOSUB1220
1070 X=X+50:GOSUB1230
1080 X=X+20:GOSUB1200
1090 X=X+50:GOSUB1190
1100 X=X+50:GOSUB1240
1110 X=X+50:GOSUB1250
1120 POKE65496,0
1130 FORS1=0TO1
1140 IFS1=1THENPALETTE0,0
1150 FORP=1TO20:FORS=S1 TO15:PAL
ETTES,RND(63):NEXT:NEXT:NEXT:PAL
ETTE0,0
1160 GOTO2000

```

```

1170 GOSUB13:X$=X$+"R40D10L30D50
R20U20L10U10R20D40L40U70":HDRAWX
$:GOSUB12:RETURN
1180 GOSUB13:X$=X$+"D70R10U30M+2
0,+30R10M-20,-30R20U40L40EM+10,+
10D20R20U20L20":HDRAWX$:GOSUB12:
RETURN
1190 GOSUB13:X$=X$+"D70R10U30R20
D30R10U70L40EM+10,+10D20R20U20L2
0":HDRAWX$:GOSUB12:RETURN
1200 GOSUB13:X$=X$+"D70R10U50M+2
0,+50R10U70L10D50M-20,-50L10":HD
RAWX$:GOSUB12:RETURN
1210 GOSUB13:X$=X$+"D70R20E20U30
H20L20EM+10,+10D50R7E14U22H14L7"
:HDRAWX$:GOSUB12:RETURN
1220 GOSUB13:X$=X$+"D70R10U30R20
U10L20U20R30U10L40":HDRAWX$:GOSU
B12:RETURN
1230 GOSUB13:X$=X$+"D70R10U70L10
":HDRAWX$:GOSUB12:RETURN
1240 GOSUB13:X$=X$+"D70R40U10L30
U60L10":HDRAWX$:GOSUB12:RETURN
1250 GOSUB13:X$=X$+"D70R40U10L30
U20R20U10L20U20R30U10L40":HDRAWX
$:GOSUB12:RETURN
2000 POKE65497,0:HSCREEN2
2010 FORS=0TO15:READA:PALETTES,A
:NEXT:DATA0,36,18,6,46,40,48,52,
27,13,2,51,50,35,31,63
2020 HCOLOR11:FORX=0TO300STEP20:
FORY=130TO180STEP10:HLINE(X,Y)-(
X+20,Y+10),PSET:HLINE(X+20,Y)-(X
,Y+10),PSET:NEXT:NEXT:HLINE(0,13
0)-319,191),PSET,B: C=8
2030 FORX=0TO300STEP20:FORY=130T
O180STEP10:HPAINT(X+5,Y+5),C,11:
IFC=8THENC=9ELSEC=8
2040 HPAINT(X+5,Y+1),C,11:IFC=8T
HENC=9ELSEC=8
2050 NEXT:NEXT
2060 FORX=10TO315STEP20:HPAINT(X
,188),9,11:NEXT
2070 FORY=135TO185STEP10:HPAINT(
315,Y),8,11:NEXT
2080 HDRAW"COBMO,190E60L60D60":H
PAINT(5,135),0,0
2090 HCOLOR11:HLINE(0,190)-(60,1
30),PSET
2100 Y=190:FORY=0TO60STEP5:HLINE
(X,INT((190-Y)/2))-(X,Y),PSET:Y=
Y-5:NEXT:HLINE(0,0)-(60,30),PSET
2110 C=1:FORY=3TO60STEP5:HPAINT(
X,96),C,11:C=C+1:IFC>15THENC=1
2120 NEXT
2130 HLINE(0,0)-(60,30),PSET
2140 HLINE-(319,30),PSET:HLINE-(
319,0),PSET:HLINE-(0,0),PSET:HPA
INT(50,1),11,11:'CEILING
2150 HPAINT(90,31),6,11:'BACKWAL
L
2160 HCOLOR0:HDRAW"BM5,30M+50,+1
5D50M-50,+20U85":HPAINT(10,60),8
,0:'WINDOW
2170 HCIRCLE(30,65),20,2:HPAINT(
30,65),2,2:'TREE
2180 HCOLOR3:HDRAW"BM27,85D20M+6
,-2U18L6":HPAINT(30,90),3,3:'TRU
NK
2190 HDRAW"C4BM50,150M+70,-50;R1
80G50L200":TABLE
2200 HPAINT(70,140),4,4
2210 HDRAW"C5BM50,150D10R200U10L
200":HPAINT(60,154),5,5
2220 HDRAW"C5BM250,150E50D10G50U
10":HPAINT(265,140),5,5
2230 HDRAW"C3BM50,160D30R5U30L5"

```

```

:HPAINT(52,162),3,3:'LEGS
2240 HDRAW"BM245,160D30R5U30L5":
HPAINT(247,162),3,3
2250 HDRAW"BM300,110G5D25R5U30":
HPAINT(298,115),3,3
2260 FORX=110TO260STEP50:FORY=12
0TO140STEP20:HCIRCLE(X-(Y-120),Y
),20,15,.3:HPAINT(X-(Y-120),Y),1
5,15:HCIRCLE(X-(Y-120),Y),10,1,.
3:HPAINT(X-(Y-120),Y),1,1:NEXT:N
EXT:'PLATES
2270 FORY=110TO260STEP45:HDRAW"C
7BM"+STR$(X)+"",110M+20,-40M+20,+
40":HCIRCLE(X+20,110),20,..2,0,.
5:HPAINT(X+20,110),7,7:HCIRCLE(X
+20,72),4,1:HPAINT(X+20,72),1,1:
HCIRCLE(X+20,90),5,9:HPAINT(X+20
,90),9,9:NEXT:'PARTY HATS
2280 HDRAW"S8C6BM260,190U10E2U4R
2D4F2D10L6":HPAINT(262,185),6,6:
HCIRCLE(266,175),6,9:HDRAW"C9S4B
M265,178U2NR2U3R4":'STUBBY
2290 HDRAW"BM160,10C1G15R30H15":
HPAINT(160,20),1,1:HLINE(158,5)-
(162,15),PSET,BF:'LIGHT
2295 HCOLOR9:HLINE(70,45)-(290,6
6),PSET,BF
2300 HCOLOR11:HPRINT(13,6),"LET'
S HAVE A PARTY!!"
2310 X=10:Y=7:GOSUB10
2330 HLINE(70,45)-(180,35),PSET
2340 HLINE-(290,45),PSET:HLINE(1
80,35)-(182,32),PRESET
2350 IS=INKEY$:GOSUB5
2360 WIDTH40:HSCREEN0:PALETTECMP
:LOCATE10,10:ATTR2,1,B:PRINT"Any
ther go? <Y>es or <N>o";ATTR6,0
:GOSUB6:GOSUB6:GOSUB6:GOSUB6:GOS
UB6:GOSUB5:IFIS="Y"THENRUN
2370 POKE65496,0:CLS
2380 END
2390 POKE65496,0:PALETTECMP

```

HINT.....

Colours

Using the PALETTE command on the text screen. You can now change your 32x16 text screen from the normal black on green to any other colour that the CoCo 3 has.

For example, if you wanted a white script with a black background, (the colour for white is 63, the colour for black is 0) you would type in:

```

PALETTE 12,63:PALETTE13,0
'PALETTE12' controls the
colour for the script (the
lettering, etc) while
'PALETTE13' controls the
background colours.

```

The new version of....

TANK II

16K ECB
GAME

by Craig Stewart

TANK II ... MY NEW version of Tank Battle (September 1986 Aust. CoCo) - it is basically the same as the old, except that ...

1. Parameters can no longer be changed.

2. New, simplified backgrounds - on starting the game, press '1' for black with white squares, '2' for white with black squares, '3' for all black and '4' for all white.

3. Tanks have 30 bullets, but longer reloading time.

4. Both players recover from a hit at the same time.

5. A player can only conclusively win if he/she can get out of the way of the exploding bits of the opposition's tank on the final shot. A player still wins if he/she is not the first destroyed, but a conclusive win should be considered a much better win.

Also each tank has some new functions. Pulling the joystick back moves the tank forward, and deposits a 'glitter' behind it. This 'glitter' can do many things - it can chew through obstacles very quickly without having to blast through. It can sometimes help you to survive a direct hit on your tank from an enemy bullet, or make the bullet explode on the 'glitter' left behind you, and it can also obstruct the tank that is chasing you. While in the 'glitter' mode, your tank also has the power to go into 'hyperjump' - disappear, and reassemble later, in another place - this is done by pressing the button.

You only have a limited amount of 'glitter' and only 3 hyperjumps. You can not start the glitter mode if you are directly in front of an obstacle.

This update might not be everyone's cup of tea, because of its greater complexity, but you still might like to watch the special effects and listen to the sound each time a tank gets blown to bits.

The Listing:

```
1 REM CLEAR MEMORY IF NECESSARY
2 FOR I= 28672 TO 31666
3 READ QS:POKEI,VAL("&H"+QS):NEX
TI
4 EXEC 28672
5 DATA 8E,80,0,BF,1,13,8E,77,64,
10,8E,7B,B2,A6,80,A7,A0,8C,77,D2
,25,F7,7E,79,B6,7F,77,8E,17,7,B4
,8D,10,17,7,D3,7C,77,8E,17,7,BB,
8D,5,17,7,D9,20,E8,20,1F,AD,9F,A
0,A,BE,77,66,E6,84,26,4A,86,50,B
7,77,8C,7A,77,6D,26,40,B6,77,6D
6 DATA 8E,10,B7,77,6D,20,36,BE,1
,13,8C,C0,0,25,6,8E,80,0,BF,1,13
,7D,77,81,27,CE,7A,77,81,26,11,B
D,7B,8D,BD,7A,9D,86,50,B7,7B,B0,
BD,7A,BF,BD,7A,AE,8E,1,72,BD,72,
45,30,1F,26,F9,16,0,FB,C1,3F,26,
12,86,50,B7,77,8C,B6,77,6D,4C,81
7 DATA 11,25,2,86,1,B7,77,6D,17,
3,DB,8E,77,4,F6,77,6D,5A,86,6,3D
,30,8B,C6,4,F7,77,75,7A,77,75,27
,1B,B6,77,64,AB,80,B7,77,89,B6,7
7,65,AB,80,B7,77,8A,34,10,17,3,D
,35,10,26,E2,20,4A,BE,77,68,E6,8
4,27,4,C1,3F,26,2E,86,64,B7,77
8 DATA 8C,F6,77,6D,5A,58,8E,76,E
4,3A,A6,84,BB,77,64,81,1,25,17,8
1,EF,22,13,B7,77,64,A6,1,BB,77,6
5,81,1,25,7,81,AF,22,3,B7,77,65,
BE,77,68,E6,84,C1,3F,26,5,F7,77,
80,20,3,7F,77,80,F6,77,6D,8E,7A,
BB,5A,86,20,3D,30,8B,BF,77,7B,17
9 DATA 2,F5,7D,77,7D,26,13,7D,77
,80,27,E,BE,1,13,BF,77,73,7A,77,
7E,26,9,7C,77,7D,BE,77,7B,BF,77,
73,7D,77,79,26,A,7A,77,7A,26,5,8
6,1E,B7,77,79,7D,77,6B,26,C,B6,F
F,0,B4,77,6A,10,26,0,2,8D,35,C6,
5,F7,77,8B,7A,77,8B,27,D,17
10 DATA 0,7C,86,5F,BD,72,45,4A,2
6,FA,20,EE,B6,77,85,80,A,B1,77,8
C,25,4,B7,77,85,39,B6,77,8C,B7,7
7,85,81,1E,25,2,80,8,B7,77,8C,39
,7D,77,82,27,7,7D,77,80,10,26,9,
C7,7D,77,79,10,27,8,85,7A,77,79,
27,35,86,4,B7,77,76,86,FF,B7,77
11 DATA 85,86,37,B7,77,86,7C,77,
6B,B6,77,64,8B,8,B7,77,71,B6,77,
65,8B,8,B7,77,72,F6,77,6D,5A,58,
8E,76,E4,3A,A6,84,B7,77,6E,A6,1,
B7,77,6F,B6,77,92,B7,77,7A,39,7D
,77,6B,26,1,39,17,2,66,B6,77,71,
BB,77,6E,81,FE,22,35,81,1,25,31,
B7
12 DATA 77,71,B7,77,89,B6,77,72,
BB,77,6F,81,1,25,21,81,BE,22,1D,
B7,77,72,B7,77,8A,7A,77,76,26,A,
7C,77,76,17,1,A8,10,27,0,3C,17,1
```

```
,8A,7A,77,86,26,7,7F,77,6B,17,2,
21,39,39,34,2,B6,FF,23,8A,8,B7,F
F,23,FE,77,8F,33,41,11,83,A4,10
13 DATA 25,3,CE,9C,40,FF,77,8F,A
6,C4,B1,77,85,25,4,F6,77,85,3D,B
7,FF,20,35,2,39,7F,77,6B,86,FF,B
7,77,85,7D,77,8E,26,8,BD,77,F7,B
D,77,E5,20,6,BD,78,8,BD,77,D3,17
,1,ED,17,1,47,27,43,7C,77,91,B6,
77,92,44,44,B7,77,7A,BD,78,42,BD
14 DATA 79,DE,BD,78,19,7F,77,79,
7C,77,78,B6,77,77,81,C7,22,62,8B
,32,B7,77,77,BD,7A,9D,B6,77,77,C
6,1E,3D,8B,28,B7,7B,B1,B7,7B,B0,
BD,78,6B,BD,7A,AE,BD,7A,BF,BD,7A
,AE,7D,77,8E,26,8,BD,78,8,BD,77,
D3,20,6,BD,77,F7,BD,77,E5,7D,77,
91,27
15 DATA E,B6,77,92,44,44,B7,77,7
A,7F,77,91,7F,77,79,B6,77,89,80,
8,B7,77,87,B6,77,8A,80,8,B7,77,8
8,17,0,E0,10,8E,76,BB,16,1,78,17
,1,62,BD,78,42,BD,7A,8B,BD,78,19
,B6,77,64,B7,77,87,B6,77,65,B7,7
7,88,BD,78,6B,B6,77,64,B7,77,83,
B6
16 DATA 77,65,B7,77,84,7D,77,8E,
27,5,BD,77,E5,20,3,BD,77,D3,BD,7
A,9D,BD,74,23,8E,E,0,10,8E,26,0,
20,11,86,4,8E,0,0,30,1F,26,FC,4A
,26,F6,30,64,7E,79,E6,EC,81,10,A
3,A1,26,A,8C,26,0,25,F4,77,83
,20,3,7C,77,83,1A,10,7F,77,84
17 DATA C6,A,B6,77,84,4A,26,FD,7
3,FF,20,5A,26,F4,7A,77,84,26,ED,
7D,77,83,26,18,7C,77,84,C6,A,B6,
77,84,4A,26,FD,73,FF,20,5A,26,F4
,7C,77,84,26,ED,20,A6,BD,78,42,B
D,7A,88,BD,78,19,BD,78,6B,20,98,
B6,77,71,B7,77,87,B6,77,72,B7,77
,88,17
18 DATA 0,1F,B6,77,83,AA,84,A7,8
4,39,B6,77,89,B7,77,87,B6,77,8A,
B7,77,88,17,0,8,E6,84,FA,77,83,E
1,84,39,BD,72,45,B6,77,87,C6,20,
3D,B7,77,84,86,8,3D,8E,76,DB,30,
86,E6,84,F7,77,83,B6,77,88,C6,20
,3D,1F,1,F6,77,84,3A,30,89,E,0,1
0
19 DATA BE,77,7B,86,10,B7,77,6C,
39,B6,77,64,B7,77,87,B6,77,65,B7
,77,88,8D,BF,10,BE,77,7B,86,10,B
7,77,6C,BD,72,45,A6,A0,F6,77,83,
3D,AA,84,A7,80,EA,84,E7,84,A6,A0
,F6,77,83,3D,AA,84,A7,80,EA,84,E
7,84,7A,77,6C,27,5,30,88,1E,20,D
7,39,B6
20 DATA 77,71,B7,77,87,B6,77,72,
B7,77,88,17,FF,7D,B6,77,83,43,A4
,84,A7,84,39,B6,77,64,B7,77,87,B
6,77,65,B7,77,88,17,FF,65,10,BE,
77,73,A6,A0,F6,77,83,3D,43,53,A4
```

84, A7, 80, E4, 84, E7, 84, A6, A0, F6, 7
7, 83, 3D, 43, 53, A4, 84, A7, 80, E4, 84,
E7, 84, 7A
21 DATA 77, 6C, 10, 27, FF, AD, 30, 88,
1E, 20, D4, 0, 0, 0, 1, 80, 19, 98, 19, 9
8, 19, 98, 1F, F8, 1F, F8, 1F, F8, 1F, F8,
1F, F8, 1F, F8, 18, 18, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 6, 20, 6, 60, C, 60, E, CC, 1F, CC,
1F, FC, 1F, F8, 3F, F8, 37, F8, 1, F0, 0, 7
0, 0, 60, 0, 0, 0, 0
22 DATA 0, 0, 1, 80, 3, 80, 7, 10, E, 38,
1F, 70, 3F, E0, 7F, C6, 7F, EE, F, FC, 7, F
8, 3, F0, 1, E0, 1, C0, 1, 80, 0, 0, 0, 0, 0,
20, 0, E0, 7, C0, 1F, 80, 3F, 9C, 3F, F8, F
E0, F, C0, 7, EC, 7, FC, 3, F0, 7, C0, 6, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 1F, F8, 1F, F8
, F
23 DATA C0, F, C0, F, FC, F, FC, F, C0, F
, C0, 1F, F8, 1F, F8, 0, 0, 0, 0, 0, 0, 0,
0, 0, 6, 0, 7, C0, 3, F0, 7, FC, 7, EC, F, C0
, F, E0, 3F, F8, 3F, 9C, 1F, 80, 7, C0, 0, E
0, 0, 20, 0, 0, 0, 0, 1, 80, 1, C0, 1, E0, 3,
F0, 7, F8, F, FC, 7F, EE, 7F, C0, 3F, E0, 1
F, 70
24 DATA E, 38, 7, 10, 3, 80, 1, 80, 0, 0,
0, 0, 0, 0, 0, 60, 0, 70, 1, F0, 37, F8, 3F,
F8, 1F, F8, 1F, FC, 1F, CC, E, C6, C, 60, 6
, 60, 6, 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 18,
18, 1F, F8, 1F, F8, 1F, F8, 1F, F8, 1F, F8
, 1F, F8, 19, 98, 19, 98, 19, 98, 1, 80, 0,
0, 0, 0, 0
25 DATA 0, 0, 0, 6, 0, E, 0, F, 80, 1F, EC
, 1F, FC, 1F, F8, 3F, F8, 33, F8, 63, 70, 6
, 30, 6, 60, 4, 60, 0, 0, 0, 0, 0, 1, 80, 3
, 80, 7, 80, F, C0, 1F, E0, 3F, F0, 77, FE,
63, FE, 7, FC, E, F8, 1C, 70, 8, E0, 1, C0,
1, 80, 0, 0, 0, 0, 0, 0, 20, 3, E0, F, C0,
3F, E0
26 DATA 37, E0, 3, F0, 7, F0, 1F, FC, 39
, FC, 1, F8, 3, E0, 7, 0, 4, 0, 0, 0, 0, 0,
0, 0, 0, 1F, F8, 1F, F8, 3, F0, 3, F0, 3F, F
0, 3F, F0, 3, F0, 3, F0, 1F, F8, 1F, F8, 0,
0, 0, 0, 0, 0, 0, 4, 0, 7, 0, 3, E0, 1, F8,
39, FC, 1F, FC, 7, F0, 3, F0, 37, E0, 3F, E
0, F
27 DATA C0, 3, E0, 0, 60, 0, 0, 0, 0, 0,
1, 80, 1, C0, 8, E0, 1C, 70, E, F8, 7, FC,
63, FE, 77, FE, 3F, F0, 1F, E0, F, C0, 7, 8
0, 3, 80, 1, 80, 0, 0, 0, 0, 0, 4, 60, 6, 6
0, 6, 30, 63, 70, 33, F8, 3F, F8, 1F, F8, 1
F, F8, 1F, EC, F, 80, E, 0, 6, 0, 0, 0, 0, 0,
0, 0
28 DATA 0, 0, 0, 0, 0, 0, 0, 1, 80, 7, E
0, F, F0, F, F0, 7, E0, 7, C0, 1, 80, 0, 0, 0
, 0, 0, 0, 0, 80, 40, 20, 10, 8, 4, 2, 1, 0
, 0, FE, 1, FE, 2, FE, 2, FF, 2, 0, 2, 1, 2, 2
, 1, 2, 0, 2, FF, 2, FE, 2, FE, 1, FE, 0, FE,
FF, FE, FE, FF, FE, 2, 0, 8, 0

29 DATA D, 0, 5, 1, B, 1, F, 5, 7, 0, D, 2,
F, 6, A, 0, E, 4, E, 9, E, 3, E, D, E, 9, E, 4,
C, A, A, F, F, 8, D, D, 9, F, F, B, B, E, 6, E,
3, E, 7, F, C, E, 0, B, 5, F, 9, E, 0, 9, 4, D,
6, F, 1, 6, 1, B, 5, F, 1, 8, 2, 3, 2, C, 4
30 DATA 1, 1, 5, 9, 1, 0, 6, 3, 1, 6, 0, 0,
4, 4, 1, 9, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 23, 0, 0
, 9C, 40, 0, FF, 5, A4, 1, 5A, 1, 5B, 1, 0, 0
, 1, 0
31 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1E, 0
, 0, 0, 0, 0, 0, 0, 3, 0, EE, 5, 1, 5C, 1, 5
D, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1E, 0, 0, 0, 0, 0, 0, 0, 3, 0, 8E, 77, 64,
10, 8E, 77, 93, A6, A0, A7, 80, 10, 8C, 77
, B2, 25, F6, 39, 8E, 77, 64, 10
32 DATA 8E, 77, B3, A6, A0, A7, 80, 10,
8C, 77, D2, 25, F6, 39, 8E, 77, 93, 10, 8E
, 77, 64, A6, A0, A7, 80, 8C, 77, B2, 25, F
7, 39, 8E, 77, B3, 10, 8E, 77, 64, A6, A0,
A7, 80, 8C, 77, D2, 25, F7, 39, 7F, FF, C0
, 7F, FF, C3, 7F, FF, C5, B6, FF, 22, 84, 7
, 8A, F8, B7, FF, 22, 7F, FF, C7, 7F, FF, C
9, 7F, FF
33 DATA CB, 7F, FF, CC, 7F, FF, CE, 7F,
FF, DO, 7F, FF, D2, 39, 7F, FF, C0, 7F, FF
, C3, 7F, FF, C5, B6, FF, 22, 84, 7, 8A, E8
, B7, FF, 22, 7F, FF, C7, 7F, FF, C9, 7F, F
F, CB, 7F, FF, CC, 7F, FF, CE, 7F, FF, D0,
7F, FF, D2, 39, B6, 77, 64, B7, 77, 87, B6
, 77, 65, B7, 77, 88, 8E, 79, 81, 10, 8E, 7
9, A1, A6
34 DATA 80, A7, A0, 10, 8C, 79, C1, 25,
F6, 8E, 79, 61, B6, 77, 64, 8B, 8, F6, 77,
65, CB, 8, A7, 80, E7, 80, 8C, 79, 81, 26,
F7, 8E, 79, 61, 10, 8E, 79, 81, 4F, C6, FE
, F7, 77, 85, E6, 86, E0, A6, E0, A6, E0, A
6, F7, 77, 64, 4C, E6, 86, E0, A6, E0, A6,
E0, A6, F7, 77, 65, 17, 0, 4D, 4A, E6, 86,
EB, A6
35 DATA F7, 77, 64, E7, 86, 4C, E6, 86,
EB, A6, F7, 77, 65, E7, 86, 17, 0, 18, 4C,
81, 1F, 25, CA, 4F, 7A, 77, 85, 7A, 77, 85
, 7A, 77, 85, 7A, 7B, B1, 10, 27, 0, CF, 20
, B7, 17, F9, 4E, 34, 16, B6, 77, 65, 81, B
F, 25, 2, 86, BF, B7, 77, 65, 17, 0, 29, B6
, 77, 83, AA, 84, A7, 84, 35, 16, 39, 17, F
9, 30
36 DATA 34, 16, B6, 77, 65, 81, BF, 25,
2, 86, BF, B7, 77, 65, 17, 0, B, B6, 77, 83
, 43, A4, 84, A7, 84, 35, 16, 39, B6, 77, 6
4, C6, 20, 3D, B7, 77, 84, 86, 8, 3D, 8E, 7
6, DB, 30, 86, E6, 84, F7, 77, 83, B6, 77,
65, C6, 20, 3D, 1F, 1, F6, 77, 84, 3A, 30,
89, E, 0, 39, 80, 40, 20, 10, 8, 4, 2, 1
37 DATA 0, 80, 60, 80, 60, 80, 60, 80, 6
0, 80, 60, 80, 60, 80, 60, 80, 60, 80, 60,

80, 60, 80, 60, 80, 60, 80, 60, 80, 60, 80
, 60, 80, 60, 0, 1, 1, 1, FF, 1, FF, FF, 1, 0
, 1, FF, 0, FF, FF, 0, 1, FE, 2, FF, 2, 1, 1,
2, FF, 2, FE, 1, FE, FF, FF, FE, 0, 0, 0, 0,
0, 0, 0, 0, 0
38 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 8E, 79, A1, 10,
8E, 79, 81, A6, 80, A7, A0, 8C, 79, C1, 25
, F7, B6, 77, 87, B7, 77, 64, B6, 77, 88, B
7, 77, 65, 39, 8E, 11, 94, 30, 1F, 26, FC,
39, 1C, EF, BD, 78, 19, 10, 8E, 7B, B2, 8E
, 77, 64, A6, A0, A7, 80
39 DATA 8C, 77, D2, 25, F7, 17, 0, 2E, A
D, 9F, A0, 0, 27, FA, 81, 31, 26, 2, 20, 1C
, 81, 32, 26, 5, BD, 7A, 3A, 20, 13, 81, 33
, 26, 3, 7E, 70, 19, 81, 34, 26, 6, BD, 7A,
3A, 7E, 70, 19, 20, D8, 17, 0, 1F, 7E, 70,
19, 8E, E, 0, CC, 0, 0, ED, 81, 8C, 26, 0, 2
5, F9, 39, 8E, E, 0, CC, FF, FF, ED
40 DATA 81, 8C, 26, 0, 25, F9, 39, 8E, 1
0, 83, 20, 10, 86, 25, C6, 5, 63, 80, 5A, 2
6, FB, 30, 88, 1B, 4A, 26, F3, 39, 8D, EE,
30, 89, 2, 80, 8D, EB, 30, 89, 2, 80, 8D, E
2, 8E, 10, 8D, 8D, DD, 30, 89, 9, A0, 8D, D
7, 8E, 10, 97, 8D, D2, 30, 89, 2, 80, 8D, C
C, 30, 89, 2, 80, 8D, C6, 39, C6, FE, F7, 7
7
41 DATA 85, 1F, 98, 4A, 26, FD, 34, 4, B
D, 72, 45, 35, 4, 5A, 26, F1, 39, 8E, E, 0,
10, 8E, 26, 0, EC, 81, ED, A1, 8C, 26, 0, 2
5, F7, 39, 8E, E, 0, 10, 8E, 26, 0, EC, A1,
ED, 81, 8C, 26, 0, 25, F7, 39, B6, 77, 64,
B7, 77, 87, B6, 77, 65, B7, 77, 88, 8E, 79
, 81, 10, 8E, 79, A1, A6, 80, A7, A0, 10
42 DATA 8C, 79, C1, 25, F6, 8E, 79, 61,
B6, 77, 64, 8B, 8, B7, 77, 64, F6, 77, 65,
CB, 8, F7, 77, 65, A7, 80, E7, 80, 8C, 79,
81, 26, F7, 8E, 79, 81, 10, 8E, 79, 61, A6
, 80, C6, FF, F0, 7B, B0, 3D, FB, 77, 64, B
7, A0, A6, 80, C6, FF, F0, 7B, B0, 3D, FB,
77, 65, E7, A0, 10, 8C, 79, 81, 25, E0, 8E
, 79, 61
43 DATA 10, 8E, 79, 81, 4F, C6, A, F7, 7
7, 85, E6, 86, E0, A6, E0, A6, E0, A6, F7,
77, 64, 4C, E6, 86, E0, A6, E0, A6, E0, A6
, F7, 77, 65, 17, FD, CC, 4A, E6, 86, EB, A
6, F7, 77, 64, E7, 86, 4C, E6, 86, EB, A6,
F7, 77, 65, E7, 86, 17, FD, 97, 4C, 81, 1F
, 25, CA, 4F, F6, 77, 85, CB, 3, F7, 77, 85
, 7A, 7B
44 DATA B0, 10, 27, FE, 4F, 20, B8, 7A,
77, 82, BE, 77, 7B, BF, 77, 73, BD, 74, 7C
, 86, 2A, B7, 77, 81, 32, 62, 86, C7, B7, 7
7, 85, 39, BE, 1, 13, A6, 80, 81, A, 25, F7
, 81, E6, 22, F3, B7, 77, 64, A6, 80, 81, A
, 25, FA, 81, AA, 22, F6, B7, 77, 65, 86, F
F, B7, 77, 85, 39, 0, 0, 3F

DISASSEMBLER

continued from page 9

MIR BGER BLTR BGTR BLER ": RETURN
196 S\$="LEAXILEAYILEASILEAUIPSHS
MPULSMPSHUMPULUM****RTSH ABXH R
TIH CVAIHMULH *****SWIH ": RETURN
197 S\$="NEGAH*****COMAHLRSRA
H*****RORAHASRAHASLAHROLAHDCAH*
****INCAHTSTAH*****CLRAH": RETURN
198 S\$="NEGBH*****COMBHLRSRB
H*****RORBHASRBRHASLHROLBHDCEBH*
****INCBHTSTBH*****CLRBH": RETURN
199 S\$="NEGI *****COMI LSRI
*****RORI ASRI ASLI ROLI DECI *
****INCI TSTI JMPI CLRI ": RETURN

200 S\$="NEGE *****COME LSRE
*****RORE ASRE ASLE ROLE DECE *
****INCE TSTE JMPE CLRE ": RETURN
201 S\$="SUBA1CMPA1SBCA1SUBD2ANDA
1BITA1LDA1 *****EORA1ADCA1ORA1 A
DDA1CMPX2BSRR LDX2 *****": RETURN
202 S\$="SUBA CMPA SBCA SUBD ANDA
BITA LDA STA EORA ADCA ORA A
DDA CMPX JSR LDX STX ": RETURN
203 S\$="SUBB CMPB SBCB ADDD ANDB
BITB LDB STB EORB ADCB ORB A
DDB LDD STD LDU STU ": RETURN
204 S\$="SUBB1CMPB1SBCB1ADDD2ANDB
1BITB1LDB1 *****EORB1ADCB1ORB1 A
DDB1LDD2 *****LDU2 *****": RETURN
205 S\$="3FHSWI2832CMPD8C2CMPY8E2
LDY 93DCMPD9CDCMPY9EDLDY 9FDSTY
A31CMPDACCMPYAEILDY AFISTY B3EC

MPDBCECMPIYBEELDY BFESTY CE2LDS D
EDLDS DFDSTS EEILDS EFISTS FEELD
S FFSTS ": RETURN
206 S\$="3FISWI3832CMPU8C2CMP93D
CMPU9CDCMPSA3ICMPUAC1CMPSB3ECMPU
BCECMPS": RETURN
255 'HEX' _DEC ZH\$=HEX(ZN)
256 ZH\$=""
258 Z2=INT(ZN/16): Z1=ZN-Z2*16: ZH
\$=MID\$("0123456789ABCDEF", Z1+1, 1
) + ZH\$: IF Z2=0 THEN RETURN ELSE Z2:
GOTO 258
263 'INSTR\$
264 ZS=1
266 ZI=0: IF ZS<1 THEN ZS=1
268 FOR ZJ=ZS TO LEN(ZS\$): IF MID\$(
ZS\$, 1, LEN(ZT\$))=ZT\$ THEN ZI=ZJ: RE
TURN ELSE NEXT ZJ: RETURN

Sunday driving was never like this!

OBSTACLE

16K ECB
GAME

by Craig Stewart

OBSTACLE ... WRITTEN MOSTLY in BASIC, but with some small bits of M/L to speed it up, and improve appearances.

The object of the game is simple - you are driving a car (??) at the top of the screen, trying to dodge (with the left & right arrow keys) the yellow blocks that come flying at you from the bottom.

To give the player a fair chance, there is a 'force field' function (activated by pressing the space-bar), that allows the player to crash through the obstacles for a short while (activated by space-bar), that allows the player to crash through the obstacles for a short while (car will turn white when force-field is on).

You have three of these, as well as three lives. There is no limit to the number of screens (after each screen the obstacles get wider) - although I doubt if many people will make it past the 3rd screen. Score is calculated on progress throughout the screens.

You might like to put this program in high speed.

The Listing:

```
1 GOTO 10
2 **OBSTACLE**
3 ' BY CRAIG STEWART
4 SAVE"146B:3":END
5 END
10 CLEAR200,&H3F00
20 BS=CHR$(128)+CHR$(128)
30 AS=CHR$(141+16*3)+CHR$(142+16*3)
40 FS=CHR$(141+16*4)+CHR$(142+16*4)
50 CLS0:CO=&H3F00+83
60 LC=1
70 DEFUSR0=&H3F00 'SCROLL YELLOW UP
80 DEFUSR1=&H3F00+53 'INVERT
90 DEFUSR2=&H3F00+68 'SCROLL LEFT
100 FORX=&H3F00 TO&H3F00+91
110 READT$:T=VAL("&H"+T$):POKEX,T:
NEXT
120 CS=STRING$(LC,159):CLS0:TIME R=0
125 SC=SC+1
130 PT=PT+1
140 PRINT@RND(31)+440,C$;
150 Z=USR0(0)
160 P=PEEK(343):Q=PEEK(344):IFP<>255THENC=-1ELSEIFQ<>255THENC=1ELSEC=0
170 IF((A+C)<10R(A+C)>30)THENC=0
180 IFF=1THENL=L-1:IFL<1 THENF=0ELSEGOTO210
190 PRINT@A,B$;:A=A+C:PRINT@A,A$;
200 GOTO220
210 PRINT@A,B$;:A=A+C:PRINT@A,F$;
220 K$=INKEY$:IFK$="" THENN=N+1:IFN>3THENPLAY"01T100CB"ELSEF=1:L=8:PLAY"75T100ABC"
230 M=PEEK(1056+A):O=PEEK(1057+A):IFM<>128ORO<>128THENGOTO260
240 IFTIMER>2000 GOTO290
250 GOTO 130
260 IFF=1THENZ=USR1(0):PLAY"V30T255O5CDEFGAB":Z=USR1(0):PLAY"V31":GOTO130
270 'BANG
280 DE=DE+1:Z=USR1(0):PLAY"T25501CBAGD":Z=USR1(0):PLAY"T25501V20ABCBD":Z=USR1(0):PLAY"T25501V10ABCBD":Z=USR1(0):PLAY"T25501V4ABCDGEB":Z=USR1(0):PLAY"V31":IFDE=3 GOTO420ELSECLS0:PLAY"V31":GOTO130
290 POKECO,128:PLAY"T255L25501CDEFGABO2CDEFGABO3CDEFGABO4CDEFGABO5CDEFGABO6CDEFGABO7CDEFGABO8CDEFGABO9CDEFGABO10CDEFGABO11CDEFGABO12CDEFGABO13CDEFGABO14CDEFGABO15CDEFGABO16CDEFGABO17CDEFGABO18CDEFGABO19CDEFGABO20CDEFGABO21CDEFGABO22CDEFGABO23CDEFGABO24CDEFGABO25CDEFGABO26CDEFGABO27CDEFGABO28CDEFGABO29CDEFGABO30CDEFGABO31CDEFGABO32CDEFGABO33CDEFGABO34CDEFGABO35CDEFGABO36CDEFGABO37CDEFGABO38CDEFGABO39CDEFGABO40CDEFGABO41CDEFGABO42CDEFGABO43CDEFGABO44CDEFGABO45CDEFGABO46CDEFGABO47CDEFGABO48CDEFGABO49CDEFGABO50CDEFGABO51CDEFGABO52CDEFGABO53CDEFGABO54CDEFGABO55CDEFGABO56CDEFGABO57CDEFGABO58CDEFGABO59CDEFGABO60CDEFGABO61CDEFGABO62CDEFGABO63CDEFGABO64CDEFGABO65CDEFGABO66CDEFGABO67CDEFGABO68CDEFGABO69CDEFGABO70CDEFGABO71CDEFGABO72CDEFGABO73CDEFGABO74CDEFGABO75CDEFGABO76CDEFGABO77CDEFGABO78CDEFGABO79CDEFGABO80CDEFGABO81CDEFGABO82CDEFGABO83CDEFGABO84CDEFGABO85CDEFGABO86CDEFGABO87CDEFGABO88CDEFGABO89CDEFGABO90CDEFGABO91CDEFGABO92CDEFGABO93CDEFGABO94CDEFGABO95CDEFGABO96CDEFGABO97CDEFGABO98CDEFGABO99CDEFGABO100CDEFGABO101CDEFGABO102CDEFGABO103CDEFGABO104CDEFGABO105CDEFGABO106CDEFGABO107CDEFGABO108CDEFGABO109CDEFGABO110CDEFGABO111CDEFGABO112CDEFGABO113CDEFGABO114CDEFGABO115CDEFGABO116CDEFGABO117CDEFGABO118CDEFGABO119CDEFGABO120CDEFGABO121CDEFGABO122CDEFGABO123CDEFGABO124CDEFGABO125CDEFGABO126CDEFGABO127CDEFGABO128CDEFGABO129CDEFGABO130CDEFGABO131CDEFGABO132CDEFGABO133CDEFGABO134CDEFGABO135CDEFGABO136CDEFGABO137CDEFGABO138CDEFGABO139CDEFGABO140CDEFGABO141CDEFGABO142CDEFGABO143CDEFGABO144CDEFGABO145CDEFGABO146CDEFGABO147CDEFGABO148CDEFGABO149CDEFGABO150CDEFGABO151CDEFGABO152CDEFGABO153CDEFGABO154CDEFGABO155CDEFGABO156CDEFGABO157CDEFGABO158CDEFGABO159CDEFGABO160CDEFGABO161CDEFGABO162CDEFGABO163CDEFGABO164CDEFGABO165CDEFGABO166CDEFGABO167CDEFGABO168CDEFGABO169CDEFGABO170CDEFGABO171CDEFGABO172CDEFGABO173CDEFGABO174CDEFGABO175CDEFGABO176CDEFGABO177CDEFGABO178CDEFGABO179CDEFGABO180CDEFGABO181CDEFGABO182CDEFGABO183CDEFGABO184CDEFGABO185CDEFGABO186CDEFGABO187CDEFGABO188CDEFGABO189CDEFGABO190CDEFGABO191CDEFGABO192CDEFGABO193CDEFGABO194CDEFGABO195CDEFGABO196CDEFGABO197CDEFGABO198CDEFGABO199CDEFGABO200CDEFGABO201CDEFGABO202CDEFGABO203CDEFGABO204CDEFGABO205CDEFGABO206CDEFGABO207CDEFGABO208CDEFGABO209CDEFGABO210CDEFGABO211CDEFGABO212CDEFGABO213CDEFGABO214CDEFGABO215CDEFGABO216CDEFGABO217CDEFGABO218CDEFGABO219CDEFGABO220CDEFGABO221CDEFGABO222CDEFGABO223CDEFGABO224CDEFGABO225CDEFGABO226CDEFGABO227CDEFGABO228CDEFGABO229CDEFGABO230CDEFGABO231CDEFGABO232CDEFGABO233CDEFGABO234CDEFGABO235CDEFGABO236CDEFGABO237CDEFGABO238CDEFGABO239CDEFGABO240CDEFGABO241CDEFGABO242CDEFGABO243CDEFGABO244CDEFGABO245CDEFGABO246CDEFGABO247CDEFGABO248CDEFGABO249CDEFGABO250CDEFGABO251CDEFGABO252CDEFGABO253CDEFGABO254CDEFGABO255CDEFGABO256CDEFGABO257CDEFGABO258CDEFGABO259CDEFGABO260CDEFGABO261CDEFGABO262CDEFGABO263CDEFGABO264CDEFGABO265CDEFGABO266CDEFGABO267CDEFGABO268CDEFGABO269CDEFGABO270CDEFGABO271CDEFGABO272CDEFGABO273CDEFGABO274CDEFGABO275CDEFGABO276CDEFGABO277CDEFGABO278CDEFGABO279CDEFGABO280CDEFGABO281CDEFGABO282CDEFGABO283CDEFGABO284CDEFGABO285CDEFGABO286CDEFGABO287CDEFGABO288CDEFGABO289CDEFGABO290CDEFGABO291CDEFGABO292CDEFGABO293CDEFGABO294CDEFGABO295CDEFGABO296CDEFGABO297CDEFGABO298CDEFGABO299CDEFGABO300CDEFGABO301CDEFGABO302CDEFGABO303CDEFGABO304CDEFGABO305CDEFGABO306CDEFGABO307CDEFGABO308CDEFGABO309CDEFGABO310CDEFGABO311CDEFGABO312CDEFGABO313CDEFGABO314CDEFGABO315CDEFGABO316CDEFGABO317CDEFGABO318CDEFGABO319CDEFGABO320CDEFGABO321CDEFGABO322CDEFGABO323CDEFGABO324CDEFGABO325CDEFGABO326CDEFGABO327CDEFGABO328CDEFGABO329CDEFGABO330CDEFGABO331CDEFGABO332CDEFGABO333CDEFGABO334CDEFGABO335CDEFGABO336CDEFGABO337CDEFGABO338CDEFGABO339CDEFGABO340CDEFGABO341CDEFGABO342CDEFGABO343CDEFGABO344CDEFGABO345CDEFGABO346CDEFGABO347CDEFGABO348CDEFGABO349CDEFGABO350CDEFGABO351CDEFGABO352CDEFGABO353CDEFGABO354CDEFGABO355CDEFGABO356CDEFGABO357CDEFGABO358CDEFGABO359CDEFGABO360CDEFGABO361CDEFGABO362CDEFGABO363CDEFGABO364CDEFGABO365CDEFGABO366CDEFGABO367CDEFGABO368CDEFGABO369CDEFGABO370CDEFGABO371CDEFGABO372CDEFGABO373CDEFGABO374CDEFGABO375CDEFGABO376CDEFGABO377CDEFGABO378CDEFGABO379CDEFGABO380CDEFGABO381CDEFGABO382CDEFGABO383CDEFGABO384CDEFGABO385CDEFGABO386CDEFGABO387CDEFGABO388CDEFGABO389CDEFGABO390CDEFGABO391CDEFGABO392CDEFGABO393CDEFGABO394CDEFGABO395CDEFGABO396CDEFGABO397CDEFGABO398CDEFGABO399CDEFGABO400CDEFGABO401CDEFGABO402CDEFGABO403CDEFGABO404CDEFGABO405CDEFGABO406CDEFGABO407CDEFGABO408CDEFGABO409CDEFGABO410CDEFGABO411CDEFGABO412CDEFGABO413CDEFGABO414CDEFGABO415CDEFGABO416CDEFGABO417CDEFGABO418CDEFGABO419CDEFGABO420CDEFGABO421CDEFGABO422CDEFGABO423CDEFGABO424CDEFGABO425CDEFGABO426CDEFGABO427CDEFGABO428CDEFGABO429CDEFGABO430CDEFGABO431CDEFGABO432CDEFGABO433CDEFGABO434CDEFGABO435CDEFGABO436CDEFGABO437CDEFGABO438CDEFGABO439CDEFGABO440CDEFGABO441CDEFGABO442CDEFGABO443CDEFGABO444CDEFGABO445CDEFGABO446CDEFGABO447CDEFGABO448CDEFGABO449CDEFGABO450CDEFGABO451CDEFGABO452CDEFGABO453CDEFGABO454CDEFGABO455CDEFGABO456CDEFGABO457CDEFGABO458CDEFGABO459CDEFGABO460CDEFGABO461CDEFGABO462CDEFGABO463CDEFGABO464CDEFGABO465CDEFGABO466CDEFGABO467CDEFGABO468CDEFGABO469CDEFGABO470CDEFGABO471CDEFGABO472CDEFGABO473CDEFGABO474CDEFGABO475CDEFGABO476CDEFGABO477CDEFGABO478CDEFGABO479CDEFGABO480CDEFGABO481CDEFGABO482CDEFGABO483CDEFGABO484CDEFGABO485CDEFGABO486CDEFGABO487CDEFGABO488CDEFGABO489CDEFGABO490CDEFGABO491CDEFGABO492CDEFGABO493CDEFGABO494CDEFGABO495CDEFGABO496CDEFGABO497CDEFGABO498CDEFGABO499CDEFGABO500CDEFGABO501CDEFGABO502CDEFGABO503CDEFGABO504CDEFGABO505CDEFGABO506CDEFGABO507CDEFGABO508CDEFGABO509CDEFGABO510CDEFGABO511CDEFGABO512CDEFGABO513CDEFGABO514CDEFGABO515CDEFGABO516CDEFGABO517CDEFGABO518CDEFGABO519CDEFGABO520CDEFGABO521CDEFGABO522CDEFGABO523CDEFGABO524CDEFGABO525CDEFGABO526CDEFGABO527CDEFGABO528CDEFGABO529CDEFGABO530CDEFGABO531CDEFGABO532CDEFGABO533CDEFGABO534CDEFGABO535CDEFGABO536CDEFGABO537CDEFGABO538CDEFGABO539CDEFGABO540CDEFGABO541CDEFGABO542CDEFGABO543CDEFGABO544CDEFGABO545CDEFGABO546CDEFGABO547CDEFGABO548CDEFGABO549CDEFGABO550CDEFGABO551CDEFGABO552CDEFGABO553CDEFGABO554CDEFGABO555CDEFGABO556CDEFGABO557CDEFGABO558CDEFGABO559CDEFGABO560CDEFGABO561CDEFGABO562CDEFGABO563CDEFGABO564CDEFGABO565CDEFGABO566CDEFGABO567CDEFGABO568CDEFGABO569CDEFGABO570CDEFGABO571CDEFGABO572CDEFGABO573CDEFGABO574CDEFGABO575CDEFGABO576CDEFGABO577CDEFGABO578CDEFGABO579CDEFGABO580CDEFGABO581CDEFGABO582CDEFGABO583CDEFGABO584CDEFGABO585CDEFGABO586CDEFGABO587CDEFGABO588CDEFGABO589CDEFGABO590CDEFGABO591CDEFGABO592CDEFGABO593CDEFGABO594CDEFGABO595CDEFGABO596CDEFGABO597CDEFGABO598CDEFGABO599CDEFGABO600CDEFGABO601CDEFGABO602CDEFGABO603CDEFGABO604CDEFGABO605CDEFGABO606CDEFGABO607CDEFGABO608CDEFGABO609CDEFGABO610CDEFGABO611CDEFGABO612CDEFGABO613CDEFGABO614CDEFGABO615CDEFGABO616CDEFGABO617CDEFGABO618CDEFGABO619CDEFGABO620CDEFGABO621CDEFGABO622CDEFGABO623CDEFGABO624CDEFGABO625CDEFGABO626CDEFGABO627CDEFGABO628CDEFGABO629CDEFGABO630CDEFGABO631CDEFGABO632CDEFGABO633CDEFGABO634CDEFGABO635CDEFGABO636CDEFGABO637CDEFGABO638CDEFGABO639CDEFGABO640CDEFGABO641CDEFGABO642CDEFGABO643CDEFGABO644CDEFGABO645CDEFGABO646CDEFGABO647CDEFGABO648CDEFGABO649CDEFGABO650CDEFGABO651CDEFGABO652CDEFGABO653CDEFGABO654CDEFGABO655CDEFGABO656CDEFGABO657CDEFGABO658CDEFGABO659CDEFGABO660CDEFGABO661CDEFGABO662CDEFGABO663CDEFGABO664CDEFGABO665CDEFGABO666CDEFGABO667CDEFGABO668CDEFGABO669CDEFGABO670CDEFGABO671CDEFGABO672CDEFGABO673CDEFGABO674CDEFGABO675CDEFGABO676CDEFGABO677CDEFGABO678CDEFGABO679CDEFGABO680CDEFGABO681CDEFGABO682CDEFGABO683CDEFGABO684CDEFGABO685CDEFGABO686CDEFGABO687CDEFGABO688CDEFGABO689CDEFGABO690CDEFGABO691CDEFGABO692CDEFGABO693CDEFGABO694CDEFGABO695CDEFGABO696CDEFGABO697CDEFGABO698CDEFGABO699CDEFGABO700CDEFGABO701CDEFGABO702CDEFGABO703CDEFGABO704CDEFGABO705CDEFGABO706CDEFGABO707CDEFGABO708CDEFGABO709CDEFGABO710CDEFGABO711CDEFGABO712CDEFGABO713CDEFGABO714CDEFGABO715CDEFGABO716CDEFGABO717CDEFGABO718CDEFGABO719CDEFGABO720CDEFGABO721CDEFGABO722CDEFGABO723CDEFGABO724CDEFGABO725CDEFGABO726CDEFGABO727CDEFGABO728CDEFGABO729CDEFGABO730CDEFGABO731CDEFGABO732CDEFGABO733CDEFGABO734CDEFGABO735CDEFGABO736CDEFGABO737CDEFGABO738CDEFGABO739CDEFGABO740CDEFGABO741CDEFGABO742CDEFGABO743CDEFGABO744CDEFGABO745CDEFGABO746CDEFGABO747CDEFGABO748CDEFGABO749CDEFGABO750CDEFGABO751CDEFGABO752CDEFGABO753CDEFGABO754CDEFGABO755CDEFGABO756CDEFGABO757CDEFGABO758CDEFGABO759CDEFGABO760CDEFGABO761CDEFGABO762CDEFGABO763CDEFGABO764CDEFGABO765CDEFGABO766CDEFGABO767CDEFGABO768CDEFGABO769CDEFGABO770CDEFGABO771CDEFGABO772CDEFGABO773CDEFGABO774CDEFGABO775CDEFGABO776CDEFGABO777CDEFGABO778CDEFGABO779CDEFGABO780CDEFGABO781CDEFGABO782CDEFGABO783CDEFGABO784CDEFGABO785CDEFGABO786CDEFGABO787CDEFGABO788CDEFGABO789CDEFGABO790CDEFGABO791CDEFGABO792CDEFGABO793CDEFGABO794CDEFGABO795CDEFGABO796CDEFGABO797CDEFGABO798CDEFGABO799CDEFGABO800CDEFGABO801CDEFGABO802CDEFGABO803CDEFGABO804CDEFGABO805CDEFGABO806CDEFGABO807CDEFGABO808CDEFGABO809CDEFGABO810CDEFGABO811CDEFGABO812CDEFGABO813CDEFGABO814CDEFGABO815CDEFGABO816CDEFGABO817CDEFGABO818CDEFGABO819CDEFGABO820CDEFGABO821CDEFGABO822CDEFGABO823CDEFGABO824CDEFGABO825CDEFGABO826CDEFGABO827CDEFGABO828CDEFGABO829CDEFGABO830CDEFGABO831CDEFGABO832CDEFGABO833CDEFGABO834CDEFGABO835CDEFGABO836CDEFGABO837CDEFGABO838CDEFGABO839CDEFGABO840CDEFGABO841CDEFGABO842CDEFGABO843CDEFGABO844CDEFGABO845CDEFGABO846CDEFGABO847CDEFGABO848CDEFGABO849CDEFGABO850CDEFGABO851CDEFGABO852CDEFGABO853CDEFGABO854CDEFGABO855CDEFGABO856CDEFGABO857CDEFGABO858CDEFGABO859CDEFGABO860CDEFGABO861CDEFGABO862CDEFGABO863CDEFGABO864CDEFGABO865CDEFGABO866CDEFGABO867CDEFGABO868CDEFGABO869CDEFGABO870CDEFGABO871CDEFGABO872CDEFGABO873CDEFGABO874CDEFGABO875CDEFGABO876CDEFGABO877CDEFGABO878CDEFGABO879CDEFGABO880CDEFGABO881CDEFGABO882CDEFGABO883CDEFGABO884CDEFGABO885CDEFGABO886CDEFGABO887CDEFGABO888CDEFGABO889CDEFGABO890CDEFGABO891CDEFGABO892CDEFGABO893CDEFGABO894CDEFGABO895CDEFGABO896CDEFGABO897CDEFGABO898CDEFGABO899CDEFGABO900CDEFGABO901CDEFGABO902CDEFGABO903CDEFGABO904CDEFGABO905CDEFGABO906CDEFGABO907CDEFGABO908CDEFGABO909CDEFGABO910CDEFGABO911CDEFGABO912CDEFGABO913CDEFGABO914CDEFGABO915CDEFGABO916CDEFGABO917CDEFGABO918CDEFGABO919CDEFGABO920CDEFGABO921CDEFGABO922CDEFGABO923CDEFGABO924CDEFGABO925CDEFGABO926CDEFGABO927CDEFGABO928CDEFGABO929CDEFGABO930CDEFGABO931CDEFGABO932CDEFGABO933CDEFGABO934CDEFGABO935CDEFGABO936CDEFGABO937CDEFGABO938CDEFGABO939CDEFGABO940CDEFGABO941CDEFGABO942CDEFGABO943CDEFGABO944CDEFGABO945CDEFGABO946CDEFGABO947CDEFGABO948CDEFGABO949CDEFGABO950CDEFGABO951CDEFGABO952CDEFGABO953CDEFGABO954CDEFGABO955CDEFGABO956CDEFGABO957CDEFGABO958CDEFGABO959CDEFGABO960CDEFGABO961CDEFGABO962CDEFGABO963CDEFGABO964CDEFGABO965CDEFGABO966CDEFGABO967CDEFGABO968CDEFGABO969CDEFGABO970CDEFGABO971CDEFGABO972CDEFGABO973CDEFGABO974CDEFGABO975CDEFGABO976CDEFGABO977CDEFGABO978CDEFGABO979CDEFGABO980CDEFGABO981CDEFGABO982CDEFGABO983CDEFGABO984CDEFGABO985CDEFGABO986CDEFGABO987CDEFGABO988CDEFGABO989CDEFGABO990CDEFGABO991CDEFGABO992CDEFGABO993CDEFGABO994CDEFGABO995CDEFGABO996CDEFGABO997CDEFGABO998CDEFGABO999CDEFGABO1000CDEFGABO1001CDEFGABO1002CDEFGABO1003CDEFGABO1004CDEFGABO1005CDEFGABO1006CDEFGABO1007CDEFGABO1008CDEFGABO1009CDEFGABO1010CDEFGABO1011CDEFGABO1012CDEFGABO1013CDEFGABO1014CDEFGABO1015CDEFGABO1016CDEFGABO1017CDEFGABO1018CDEFGABO1019CDEFGABO1020CDEFGABO1021CDEFGABO1022CDEFGABO1023CDEFGABO1024CDEFGABO1025CDEFGABO1026CDEFGABO1027CDEFGABO1028CDEFGABO1029CDEFGABO1030CDEFGABO1031CDEFGABO1032CDEFGABO1033CDEFGABO1034CDEFGABO1035CDEFGABO1036CDEFGABO1037CDEFGABO1038CDEFGABO1039CDEFGABO1040CDEFGABO1041CDEFGABO1042CDEFGABO1043CDEFGABO1044CDEFGABO1045CDEFGABO1046CDEFGABO1047CDEFGABO1048CDEFGABO1049CDEFGABO1050CDEFGABO1051CDEFGABO1052CDEFGABO1053CDEFGABO1054CDEFGABO1055CDEFGABO1056CDEFGABO1057CDEFGABO1058CDEFGABO1059CDEFGABO1060CDEFGABO1061CDEFGABO1062CDEFGABO1063CDEFGABO1064CDEFGABO1065CDEFGABO1066CDEFGABO1067CDEFGABO1068CDEFGABO1069CDEFGABO1070CDEFGABO1071CDEFGABO1072CDEFGABO1073CDEFGABO1074CDEFGABO1075CDEFGABO1076CDEFGABO1077CDEFGABO1078CDEFGABO1079CDEFGABO1080CDEFGABO1081CDEFGABO1082CDEFGABO1083CDEFGABO1084CDEFGABO1085CDEFGABO1086CDEFGABO1087CDEFGABO1088CDEFGABO1089CDEFGABO1090CDEFGABO1091CDEFGABO1092CDEFGABO1093CDEFGABO1094CDEFGABO1095CDEFGABO1096CDEFGABO1097CDEFGABO1098CDEFGABO1099CDEFGABO1100CDEFGABO1101CDEFGABO1102CDEFGABO1103CDEFGABO1104CDEFGABO1105CDEFGABO1106CDEFGABO1107CDEFGABO1108CDEFGABO1109CDEFGABO1110CDEFGABO1111CDEFGABO1112CDEFGABO1113CDEFGABO1114CDEFGABO1115CDEFGABO1116CDEFGABO1117CDEFGABO1118CDEFGABO1119CDEFGABO1120CDEFGABO1121CDEFGABO1122CDEFGABO1123CDEFGABO1124CDEFGABO1125CDEFGABO1126CDEFGABO1127CDEFGABO1128CDEFGABO1129CDEFGABO1130CDEFGABO1131CDEFGABO1132CDEFGABO1133CDEFGABO1134CDEFGABO1135CDEFGABO1136CDEFGABO1137CDEFGABO1138CDEFGABO1139CDEFGABO1140CDEFGABO1141CDEFGABO1142CDEFGABO1143CDEFGABO1144CDEFGABO1145CDEFGABO1146CDEFGABO1147CDEFGABO1148CDEFGABO1149CDEFGABO1150CDEFGABO1151CDEFGABO1152CDEFGABO1153CDEFGABO1154CDEFGABO1155CDEFGABO1156CDEFGABO1157CDEFGABO1158CDEFGABO1159CDEFGABO1160CDEFGABO1161CDEFGABO1162CDEFGABO1163CDEFGABO1164CDEFGABO1165CDEFGABO1166CDEFGABO1167CDEFGABO1168CDEFGABO1169CDEFGABO1170CDEFGABO1171CDEFGABO1172CDEFGABO1173CDEFGABO1174CDEFGABO1175CDEFGABO1176CDEFGABO1177CDEFGABO1178CDEFGABO1179CDEFGABO1180CDEFGABO1181CDEFGABO1182CDEFGABO1183CDEFGABO1184CDEFGABO1185CDEFGABO1186CDEFGABO1187CDEFGABO1188CDEFGABO1189CDEFGABO1190CDEFGABO1191CDEFGABO1192CDEFGABO1193CDEFGABO1194CDEFGABO1195CDEFGABO1196CDEFGABO1197CDEFGABO1198CDEFGABO1199CDEFGABO1200CDEFGABO1201CDEFGABO1202CDEFGABO1203CDEFGABO1204CDEFGABO1205CDEFGABO1206CDEFGABO1207CDEFGABO1208CDEFGABO1209CDEFGABO1210CDEFGABO1211CDEFGABO1212CDEFGABO1213CDEFGABO1214CDEFGABO1215CDEFGABO1216CDEFGABO1217CDEFGABO1218CDEFGABO1219CDEFGABO1220CDEFGABO1221CDEFGABO1222CDEFGABO1223CDEFGABO1224CDEFGABO1225CDEFGABO1226CDEFGABO1227CDEFGABO1228CDEFGABO1229CDEFGABO1230CDEFGABO1231CDEFGABO1232CDEFGABO1233CDEFGABO1234CDEFGABO1235CDEFGABO1236CDEFGABO1237CDEFGABO1238CDEFGABO1239CDEFGABO1240CDEFGABO1241CDEFGABO1242CDEFGABO1243CDEFGABO1244CDEFGABO1245CDEFGABO1246CDEFGABO1247CDEFGABO1248CDEFGABO1249CDEFGABO1250CDEFGABO1251CDEFGABO1252CDEFGABO1253CDEFGABO1254CDEFGABO1255CDEFGABO1256CDEFGABO1257CDEFGABO1258CDEFGABO1259CDEFGABO1260CDEFGABO1261CDEFGABO1262CDEFGABO1263CDEFGABO1264CDEFGABO1265CDEFGABO1266CDEFGABO1267CDEFGABO1268CDEFGABO1269CDEFGABO1270CDEFGABO1271CDEFGABO1272CDEFGABO1273CDEFGABO1274CDEFGABO1275CDEFGABO1276CDEFGABO1277CDEFGABO1278CDEFGABO1279CDEFGABO1280CDEFGABO1281CDEFGABO1282CDEFGABO1283CDEFGABO1284CDEFGABO1285CDEFGABO1286CDEFGABO1287CDEFGABO1288CDEFGABO1289CDEFGABO1290CDEFGABO1291CDEFGABO1292CDEFGABO1293CDEFGABO1294CDEFGABO1295CDEFGABO1296CDEFGABO1297CDEFGABO1298CDEFGABO1299CDEFGABO1300CDEFGABO1301CDEFGABO1302CDEFGABO1303CDEFGABO1304CDEFGABO1305CDEFGABO1306CDEFGABO1307CDEFGABO1308CDEFGABO1309CDEFGABO1310CDEFGABO1311CDEFGABO1312CDEFGABO1313CDEFGABO1314CDEFGABO1315CDEFGABO1316CDEFGABO1317CDEFGABO1318CDEFGABO1319CDEFGABO1320CDEFGABO1321CDEFGABO1322CDEFGABO1323CDEFGABO1324CDEFGABO1325CDEFGABO1326CDEFGABO1327CDEFGABO1328CDEFGABO1329CDEFGABO1330CDEFGABO1331CDEFGABO1332CDEFGABO1333CDEFGABO1334CDEFGABO1335CDEFGABO1336CDEFGABO1337CDEFGABO1338CDEFGABO1339CDEFGABO1340CDEFGABO1341CDEFGABO1342CDEFGABO1343C
```

OS9 POWER COMPUTING

by Fred Bisseling

With the release of the CoCo 3 and its larger tracts of memory (only accessible by OS-9), OS9 has come of age.

Well, what is OS9? It is a very powerful and flexible Multi-level, Multi-tasking, device-independent operating System. Let's break this description down to a more manageable and meaningful terminology.

Operating System- is simply a management system which controls all the devices available to you in a controlled environment.

Multi-level- as with most operating systems, OS9 allows the storage of information on disk in a structured filing system. It allows you to create files with almost any number of levels, each with its own directory (index) of the data contained therein.

Multi-user- this means that more than one person can use the system at the same time. This is only limited by the number of terminals on the system and the amount of memory required to run the tasks requested by the users.

Multi-tasking- this is the means by which two or more programs can be run at the same time. One person would for example be running a data-base from the terminal while a word-processor was being run from the COCO.

Device-independent Input/Output System- OS9 is capable of transporting data to

and from various devices in a very controlled way. OS9 expects all data to go to the "standard output path" and come from the "standard input path". This simply means that OS9 expects all "input" to come from the keyboard, and all "output" to go to the screen of the television or monitor. However, both "input" and "output" paths can be redirected to other devices.

Almost any device can be directed by OS9. A program called a 'device descriptor' (one for each device) is used to tell the computer what is available to it. Device descriptors are used to differentiate between single or double sided drives. Whether they are thirty five, forty or eighty track drives. The user is therefore able to use a variety of hardware to suit his/her needs.

WHAT DO I NEED to run OS9?

The minimum hardware requirements are:- a 64K Color Computer, a television or monitor, one disk drive.

I stress that the above items are the minimum requirements. A second disk drive would certainly be a distinct advantage. With the release of OS9 Level III, in conjunction with a COCO3 will give the user the advantage of a larger block of contiguous memory. This allows larger programs or data files to be used, alternatively a larger number of tasks (multi-tasking)

under the multi-user system. A standard COCO3 has 128K and can be expanded to 256K.

Do not be misled into believing that OS9 alone is the answer. REMEMBER, OS9 is a management system. Unless you intend to write your own software, you will still require wordprocessors or spreadsheet etc. to do the tasks you require.

WHAT SOFTWARE IS AVAILABLE to use in conjunction with OS9?

InterTAN (Tandy) have such programs as DESKMATE (June 1986 Rainbow) and BASIC09 (Dec. 1986 Rainbow). DESKMATE stands alone, and you do not need to purchase OS9 in order to use it. The Deskmate program uses OS9 in a way that is not visible to the user.

A large number of OS9 Utilities, Wordprocessors etc. are available through other software houses as are books such as The Complete Rainbow Guide To OS9 and The Official BASIC09 Tour Guide. The documentation that comes with OS9 assumes that you have a very good working knowledge of the COCO, and Disk Basic before moving up to OS9. However, books such as those just mentioned help to take you step by step through the essentials of OS9. TAKE THE PLUNGE! OS9 certainly has almost unlimited possibilities for those who just wish to enjoy the use of a very powerful operating system or the dedicated programmer.

HINT....

Next time you write a graphics program (or any program that requires a graphic display) and you need to use GET and PUT statements, don't throw your memory away!

What do I mean? Well whenever you GET something, you have to DIMension it first. So if your GET rectangle is 20*30, you'll be throwing away heaps of memory by typing in:

```
10 DIM A(20,30)
```

GET, PUT and DIM statements

Be smart! Wouldn't you save more memory if you typed in:

```
10 DIM A(17)
```

How do you figure that out? If your rectangle is 20 pixels long and 30 pixels wide, let x=20 and y=30.

Then use the following formulae:

```
A=(x*y)/n
```

'n' has a value of 37 if you're in PMODE 3 or 4, 76 if you're in PMODE 1 or 2 and 150 if you're in PMODE 0.

So, by using the above formula, one would figure it out like this:

```
A=(20*30)/37 (assuming we're in PMODE 4!)
```

```
A=600/37
```

```
A=16.216 ...
```

```
A=17
```

```
Presto!
```

Coming to grips with CoCo 3
Hirez text screen.....



COLDATUM

BUSINESS

CoCo 3 tape based system

by Clive Winsall

I'M JUST recovering from 'Future-Shock'. On Nov.14 (86) my daughter gave me a 1/4 share of the price of a CoCo 3, as a birthday present. (don't ask which!)

Well, this was the start of a new 'Love-Affair', although I WON'T part with my lovely old #1 grey lady.

Try holding down your CTRL & ALT keys (without a program in memory!), then press <Reset>. After you finish oohing and aahing press the <Reset> again! Heh Heh!

Now for the reason for writing. The competition for the programs for the CoCo 3, of course.

The 'coming to grips' with the HIRES text screen seemed difficult in the beginning, so I decided to give myself a task such as Updating my Data-Base. Well after a few(?) trying times, and five days a week since the last week in November, I have DONE IT. (There are a few line changes I am doing at the moment to update COCO-DEX to CoCo 3. It will be called Index-Datum as it's really this program.)

Color- Datum: Coco #3

This is a Data-Base for general use. Those who have either used, or read my CoCoDex Version in Aust CoCo January 1986, will recognise it, as this was a Tape-Index variation of my Master Prog. called Info-File.

The program first checks for the quantity of Low Res graphic pages PCLEAR(ed). If you type in 'POKE25,6:NEW' before loading any programs, there are no LORES graphic pages at all! This cannot be done except from the 'Command Line'! The new CoCo 3 will let you PCLEAR1 - 8 depending on the amount of memory spare, without any fuss at all.

If the above 'PCLEAR0' has been done, the program leaves it as is, using all memory possible, but if 1 to 4 pages are set, (normally 4) then this is noted, and the quantity is stored above Basics access in a

protected address as seen in line 1. At the end of a run, this condition is reset, so leaving as found.

The SP- start of prog./ EP- end of prog./ TM- Top of memory is found in line 30-35 and used to Auto SetUp the String space & File sizes to utilise all possible memory!

If you have some other use with different applications, just alter the NMS(n)- Name Strings, and the EXS(n)'s Example Strings, to more suitable ones for your needs.

Should you alter 'F' to another value the program readjusts to suit. Although watch the 'Screen-Format' to see it fits the space allocated.

I have placed plenty of REM's to help you, also the first Infopage: gives valuable data on the programs parameters, the second shows suitable entries.

By the way there is a line of 'X's of random colours to show the range of colours available for ATTR's with the selected set I have used. The first screen you see shows some text tricks on the HSCREEN2 with some mock/real Morse-code for fun although space limited me, as this is mainly for HIRES text. The 'Clear-Memory' screen is a beauty (looks graphic'y) and shows how great this CoCo is.

There is a theme of Screen treatment throughout which imparts a character to the program.

Driving the Program:

CLOAD & RUN, following the 'On-Screen' prompts, (this is not a Bare-Bones prog.) taking the time to read each Screen as you go. When at the Menu, select 'F' to make a small test File of people you know.

Each stage will lead you through easily, so relax and don't worry about mistakes as we will fix any later!

As you complete each entry, it is reviewed on screen while you do the next, allowing you to judge your methods.

Only do a few (2 or 3), then 'FIN' as requested, by the way, did you like the

Upper/lower-case help? Now we go through a File Csave to tape (as the Help Notes say) you may ignore the Recorder so as to just practice. If you like a 'Data-Tape' (blank) could be used to save it.

At the completion of File-Save like with all functions you get an Option to retain the file in memory for Reading or Printing/ Mail Labels or as we'll do now - Alter a File.

Select 'A' and keep/ change the File number as offered, then read the screen for a moment, noting the choices available to you, 'R' is add a new Record/ Entry 'O' will select a search for an Old-Record (existing) which can be 'A' ltered or 'C' ompletely replaced/ Re-done as needed.

Whatever you do, this first Menu is returned to before ending the function, giving great flexibility to editing.

All of this was done with ease due to the 'On Screen' help offered. Next we 'E'nd & a new screen shows what type of (Qty) changes occurred. After this we do another 'save', IF changes did occur!

At last, let's read our file. Try using the two character Catg-Code, 5 Char. Catg. then try 3 or more characters to 'Global' search (a fancy way of saying Look for all Items/ Entries/ Records) to find any which match the 'Common Element', that's what we entered.

If you use 'AME' you could find GAME/ LAMENT/ CAMEL and so on. Now the computers abilities are being used to find any information required and 'collating it' on Screen/ Printer/ Mailing-Labels as you require. After FIN & Keep in memory we return to the 'Main-Menu'.

All you have done up till now, is all you need to know! Every function is the same to use. The Mail-Label allows us to do 'A' ll entries or using the Read function 'S' ome only as wanted, with a message line you can change in between each

(every?) label/ batch of labels.

The 'C'opy function is a fast way of copying your files to another tape when re-organising files.

Until we find out how to utilise more memory, there is a restriction on the amount of entries & files, although fifty Records in 25 Files gives us 1250 Items stored, not bad considering the drain on memory the 'LOCATE:ATTR' commands incur!

I personally prefer a program which won't CRASH and also looks good as well as being helpful in use.

You may write to me if you have any queries/ comments as I like doing what little I can for other Coco users.

PS: Do be careful when typing in this program as I had to use some very long lines to save memory. This leads to possible problems when loaded & edited by a Word Processor for publication. Sometimes line numbers are included in previous lines, other times because a long 'Compacted' lines only 'space' is just after the 'Line-Number', this is where the line is divided.

Look for Lines without numbers and watch for missing lines. All of this is because of the amount of characters in a line as it's 'printed' to tape. The program tested by the magazine is not the printed version of course.

I have allowed for this in the 'Error-Traps'!!

The Listing:

```
0 ON ERR GOTO10000:'* Coco # 3
1 CLEAR2,32767: CLEAR200: TM=PEEK(
39)*256ORPEEK(40): SP=PEEK(25)*25
6ORPEEK(26): GP=(SP-1537)/1536: PO
KE32767, GP: IFPEEK(32767) THENPCLE
AR1
2 WIDTH40: GOTO10' *GOTO/RUN3 CSAV
Es
3 POKE65496, 0: N$="COLDATUM": CSAV
E N$: PRINTN$;" CSAVE'd now": STOP
5 '*Remove only REM-Lines ending
- 9 (ie: 199/1999, or '* PLEASE
8 'DATA-BASE. F=Fields, K=Files-Qt
y, MR=Max-Rec's, CR=No.-Chr's a Re
cord, Prog Auto-Allocates Space/F
ile. Do NOT ALTER-'3000' in lines
30/35!
9 'Alter DATA lines for other a
pplications- Records, Books, Club-
Register?
10 'Coco+File V1.8/ Coco+Dex V1.
9/ Color-Datum Vers.3.0 1981/7 T
ape!
13 '*lbox2200-INFOFILE V1.0(Mast
er)/Inventory-Logic [Lgc-Blk.V1.
5]/Cmd64-V2=INFO-COLOR [Simons-B
asl ALL Disk
14 '-The Information Collater-
```

```
15 '(C)Copyright Clive Winsall
16 - Aspendale Vic. 3195 -
17 'Dev'd-1981/7 For Private use
ONLY!
18 'COMMERCIAL' USERS WILL BE PR
OSECUTED with Utmost-VIGOUR!!
20 'Club use OK. Changing the Na
me or removing my ID's will be i
nterpeted as an- Copy-Right Infr
ingement!
23 PALETTE: PALETTE5, 25: X=RND(
-TIMER): POKE65469, 25' *Blue/Wht A
TTR5, 4 & Blue B/Gnd L/Res Txt
24 POKE65497, 1: HSCREEN2: HCOLOR3,
11: HCLS: HPRINT(8, 8), "Initialisin
g Program": GOTO30
25 HSCREEN2: IFGP=0 THEN RETURN ELSE
CLS6: HLINE(20, 50)-(300, 142), PSET
, B: HPRINT(6, 2), HD$: DU$="Informat
ion Collating Program": HCOLOR6, 1
1: FORX=1 TO LEN(DU$): HPRINT(5+X, 8)
, MID$(DU$, X, 1): SOUND230, (RND(2))
: NEXT: HPRINT(18, 12), "(c)": HCOLOR
5, 11: CLS6: DU$="Clive Winsall
26 FORX=1 TO LEN(DU$): HPRINT(12+X,
16), MID$(DU$, X, 1): SOUND220, (RND(
3)): NEXT: Z=2: DU$="-end-": FORX=1 T
O5: SOUND255, 1: HPRINT(16+X, 20), MI
D$(DU$, X, 1): IFINT(X/2)=X/2 THEN Z=
Z+1: SOUND200, Z: NEXT ELSE Z=Z-1: SOU
ND200, Z: NEXT
27 HCOLOR3, 11: HPRINT(15, 22), "(
A R )": FORX=1 TO (TL/7): NEXT: GP=0
: RETURN
29 'Tape/Disk D= -1/1(Disk disab
le Tape-Rtns adjust"1537"-line 1
?
30 TM=PEEK(39)*256ORPEEK(40): CLE
AR(TM-PEEK(27)*256ORPEEK(28))-3
000: SP=PEEK(25)*256ORPEEK(26): E
P=PEEK(27)*256ORPEEK(28): TM=PEEK
(39)*256ORPEEK(40)
35 F=6: CR=(F+2)*10: PG=EP-SP: CL=(
TM-EP)-3000: FM=TM-EP: FI=INT(CL*(
CR/100)): MR=INT(FI/CR): K=25
40 X=0: I=1: Z=0: SW=39: W=SW: TW=SW:
C=0: RC=0: V=0: L=0: Y=0: DV=0: D=-1: B
D=88: M=0: LN=0: N=0: CO=0: RN=0: CC=0
: PT=0: GP=PEEK(32767)
50 P=0: PK=1: T=0: TT=0: ZX=0: XX=0: Z
Z=0: LB=0: ST=2: TB=0: TP=1: PF=0: RC=
0: A=0: L1=0: L2=0
60 TL=7000: ER=0: EL=0: ON ERR GOTO
10000
70 DEF FNC(Y)=PEEK(&HFF22) AND 1' *
An Logic test. TRUE=Printer Off-L
ine
80 LT$=STRING$(12, 46): S$=STRING$(
SW, 32): LP$=STRING$(W, 241)' *241=
DMP105 lines
90 HD$=" * C o l o r"+CHR$(126)+
" D a t u m * "
100 G$="Press.....": CES=" Co
mm Elem.": GL$=LT$+"": VT$=" -W
ait For Instructions- "
110 MS$="Message: (Mail-Label Uti
lity Line#110)": NAMES="Clive Win
sall.": PC$=" Vic. 3195": ID$="Sen
der": "+N$+FC$:" *Change to yours
! Make' ID$' Max 32 Chr
120 F$(0)=" * Options ": F$(1)
=" * Loading ": F$(2)=" *Save(in
g ": F$(3)=" *New-File ": F$(4)=
" * Reading ": F$(5)=" *Altering
": F$(8)=" *Hard-copy ": F$(9)="
*Mail-Labels : "
130 DIM N(K), Q$, LNS, N$, W$, BD$, DU
```

```
$, P$(F), FL$(K), NMS(F), A$(MR, F)
139 'Start Program-flow
140 GOSUB220: GOSUB270: GOSUB280: P
ALETTE0, 14: GOSUB25: GOTO1600
149 'Dump Old-Keys, U/Case
150 LOCATE7, 23: ATTR7, 4: PRINT" Ne
w-R/Old/ "": ATTR2, 4: PRINT"END-(R
/O/": ATTR3, 7, B: PRINT"E";
160 ATTR2, 4: PRINT") "": LOCATE19,
20
170 DU$=INKEY$: POKE282, 255
180 Q$=INKEY$: IFQ$="" GOTO180
190 SOUND200, 1: RETURN
199 'DATA, LEN(NMS)=10!
200 DATA "Name(s) ", "Address
", "City P/C. ", "P-B Phone "
210 DATA "Category ", "Catg.Code
"
220 FORX=1 TO F: READN$(X): NEXT: RE
TURN
250 DATA "DR. John G. SMYTHE", "16A
Collins St.", "MELBOURNE VIC. 30
00", "03/1234567- Same"
260 DATA "MEDI (Doc/Chem/Pod)", "
Use 2 max. M/M1/M2 etc"
270 FORX=1 TO F: READEX$(X): NEXT: RE
STORE: RETURN
280 FORX=1 TO K: FL$(X)="DATUM"+MI
D$(STR$(X), 2)
282 IFLEN(FL$(X)) < 8 THEN DU$="0"+R
IGHT$(FL$(X), 1): FL$(X)=LEFT$(FL$(
X), 6)+DU$: GOTO285
284 IFLEN(FL$(X)) > 8 THEN DU$=RIGHT
$(FL$(X), 3): FL$(X)=LEFT$(FL$(X),
4)+MID$(FL$(X), 5, 2)+DU$
285 NEXT: RETURN
289 'Hard-Cpy end
290 IFY THEN PRINT#DV, LP$: CHR$(27
)+CHR$(19)' *Normal text code
295 W=SW: WIDTHW+1: RETURN
299 'Screen-Header/Page-Fmt
300 CLS6: LOCATE((SW+1)/2-(LEN(HD
$)/2), 0): ATTR3, 0: PRINTHD$: ATTR2
, 5: PRINT" "
305 LOCATE0, 1: ATTR7, 4, U: PRINTN$;
: LOCATE0, 2: ATTR2, 4: PRINTN$; : LOCA
TE0, 2: RETURN
309 'Printer Header-Rtn
310 PRINT#DV, TAB(ST)"File No.": :
PRINT#DV, USING"###"; N: : PRINT#DV,
TAB((W/2)-(LEN(HD$)/2))HD$:
315 PRINT#DV, TAB(W-10)FL$(N); CHR
$(13); LP$
320 ST=2: TB=ST: PRINT#DV, TAB(TB)N
M$(1); : FORZ=2 TO (F-1): TP=26
322 IFZ <> 4 AND W <> 132 THEN TB=POS(DV
)+(TP-(LEN(NM$(Z-1)))) ELSE IFZ=4
AND W <> 132 THEN TB=ST: PRINT#DV, CHR$(
13); ELSE IFW=132 THEN TB=POS(DV)+
(TP-(LEN(NM$(Z-1))))
324 PRINT#DV, TAB(TB)NM$(Z); : NEXT
328 PRINT#DV, TAB(W-16)RIGHT$(NM$(
F), 5); TAB(W-10)"Recd. #"; CHR$(13
); LP$: TP=0: TB=0: RETURN
329 'Print-out File
330 PT=PT+1: CC=CC+1
340 IFPT=1 GOSUB490: GOSUB480
345 IFZ <> 2 GOTO390
350 IFPT=1 THEN ATTR2, 4: PRINT"
"+F$(8)+"WIDTH"; : PRINTUSING"###"
; V: : PRINT" Baud="; BD$: : GOSUB1400
: ATTR2, 4
360 IFX < 1 GOSUB310
370 LOCATE13, 8: ATTR5, 4, U: PRINTF$(
8)+" "": ATTR2, 4: PRINT" "": LOCA
TE13, 18: ATTR2, 1: PRINT" Record #";
: PRINTUSING"###"; I: : ATTR2, 4: PRI
```

```

NT" "
380 IFXX>1ANDW<132ANDINT(XX/15)=
XX/15THENPRINT#DV,CHR$(13);CHR$(
13);CHR$(13):GOSUB310:GOTO400
385 IFXX>1ANDW=>132ANDINT(XX/30)
=XX/30THENPRINT#DV,CHR$(13);CHR$(
13);CHR$(13):GOSUB310:GOTO400
390 IFDV=0ANDPT<1ANDINT(PT/5)=PT
/5GOSUB1830:GOSUB490:GOSUB480
399 'Recd Prn-Fmt
400 IFY=OGOTO440ELSEST=2:TB=ST:T
P=26:PRINT#DV,TAB(ST)A$(I,1);:FO
RZ=2TO(F-1)
410 IFZ<4ANDW<>132THENTB=POS(DV
)+(TP-(LEN(A$(I,Z-1))))ELSEIFZ=
4ANDW<>132THENTB=ST:PRINT#DV,CHR
$(13);ELSEIFW=132THENTB=POS(DV
)+(TP-(LEN(A$(I,Z-1))))
420 PRINT#DV,TAB(TB)A$(I,Z);:NEX
TZ
430 PRINT#DV,TAB(W-16)A$(I,F);TA
B(W-10)";:PRINT#DV,USING"###";I
:GOTO450
440 ATTR2,4:PRINT" ";:HSTATD$,
A,TP,TB:PRINTA$(I,1);:LOCATE33,T
B:ATTR2,1:PRINTUSING"R###";I;A
TTR2,4:PRINT" ";:LOCATETP,TB+1:P
RINTA$(I,2);:LOCATETP,TB+2:PRINT
A$(I,3);
445 LOCATETP,TB+3:PRINTA$(I,4):L
OCATE26,TB+3:PRINTA$(I,5):LOCATE
34,TB+3:PRINTUSING"% %";"["+A$(
I,6)+"]";
450 IFY THENTB=ST ELSELOCATE0,TB
+4:ATTR5,4,U:PRINT$:ATTR2,4
455 XX=XX+1:RETURN
459 'Mail-Label Format DMP105.L1
ne 1559 shows Codes
460 PT=PT+1:CC=CC+1:CO=CO+1:IFPT
=1THENGOSUB490:GOSUB480:LOCATE4,
8:ATTR2,4:PRINT" Printer Ready t
o Start LABELS? ";:GOSUB1830:LOC
ATE0,8:ATTR2,4:PRINT$:
461 LOCATE12,8:ATTR2,4,U:PRINT$(
9);:GOSUB1400:ATTR2,4:LOCATE12,
18:ATTR2,1:PRINT" Record #";:PRI
NTUSING"###";I;:ATTR2,4:PRINT
";
462 IFDV THENPRINT#DV,CHR$(27)+C
HR$(23);CHR$(27)+CHR$(31)
463 PRINT#DV,IDS;CHR$(13)
464 IFDV THENPRINT#DV,CHR$(27)+C
HR$(32);
465 IFDV THENPRINT#DV,CHR$(27)+C
HR$(23);CHR$(27)+CHR$(14);
470 PRINT#DV,TAB(2)A$(I,1)
474 PRINT#DV,TAB(2)A$(I,2)
475 IFDV THENPRINT#DV,TAB(2)CHR$(
15);A$(I,3);CHR$(14)ELSEPRINT#D
V,A$(I,3)
476 IFDV THENPRINT#DV,CHR$(27)+C
HR$(20);CHR$(27)+CHR$(15)
477 IFDV THENPRINT#DV,CHR$(27)+C
HR$(31);TAB(1)M$;CHR$(27)+CHR$(
32);CHR$(27)+CHR$(23)
478 RETURN
479 'Menu
480 LOCATE0,2:ATTR7,4,U:PRINT$:;
:LOCATE3,2:ATTR5,4,U:PRINTCES+"(
"+SCH$+)"";:LOCATE0,3:ATTR2,4:PR
INT$;:GOSUB810:LOCATE0,3:ATTR2,
4:RETURN
490 IFY THENDV=0:TW=W:W=SW:GOSUB
300:DV=Y:W=TW:TW=SW ELSEGOSUB300
495 LP$=STRING$(W,241):RETURN
499 '*Pivot-Menu
500 POKE65497,1:PF=0:HSCREEN2:PA

```

```

LETTE5,25:GOSUB300:AUDI00N:LOCAT
E0,2:ATTR3,0
510 PRINT$(0);:ATTR2,4:PRINT" "
";:LOCATE26,2:ATTR2,1:PRINTUSING"
%###";"Max-Rec's";MR;:
ATTR2,4:PRINT" ":LOCATE0,3:GOSUB
880:ATTR2,5
520 LOCATE3,5;:PRINT"* Now Read
a File ";G$;ATTR2,4:PRINT"[R]"
";:ATTR2,5:PRINT" ":ATTR2,5,U:PRI
NT$;
525 ATTR2,5:LOCATE3,8:PRINT"* Fo
r Print-outs ";GL$;ATTR2,4:PRI
NT"[P]";:ATTR2,5:PRINT" "
527 LOCATE3,10:PRINT"* Mailing-L
abels ";GL$;ATTR2,4:PRINT"[M]"
";:ATTR2,5:PRINT" ":ATTR2,5,U:PRI
NT$;
530 ATTR2,5:LOCATE3,13:PRINT"* T
o Make a File ";GL$;ATTR2,4:PR
INT"[F]";:ATTR2,5:PRINT" "
540 LOCATE3,15;:PRINT"* Or Alter
-File ";GL$;ATTR2,4:PRINT"[A]"
";:ATTR2,5:PRINT" "
545 LOCATE3,17:IFN=0THENPRINT"*
To Copy a File ";GL$;ATTR2,4:P
RINT"[C]";:ATTR2,5:PRINT" ":GOTO
550
546 PRINT"* Save";:ATTR3,6:PRINT
" This ";:ATTR2,5:PRINT"File ";
GL$;ATTR2,4:PRINT"[C]";:ATTR2,5
:PRINT" "
550 LOCATE3,19:PRINT"* Load next
File ";GL$;ATTR2,4:PRINT"[L]"
";:ATTR2,5:PRINT" "
555 LOCATE0,20:ATTR2,0,U:PRINT$:
:LOCATE0,22:PRINT$:;LOCATE0,21
:ATTR2,4:PRINT" "
560 LOCATE3,21:PRINT"* Info-Page
s:1"+CHR$(126)+"2 *";LEFT$(GL$,1
3);"["F2]";:LOCATE2,23
570 ATTR3,7,B:PRINT" *";:ATTR3,
7:PRINT"END/ Finished. *";LEFT$(
G$,11);"["";:ATTR2,4,B:PRINT"CTRL
";:ATTR3,7:PRINT"]";
580 IFN THENLOCATE15,2:ATTR3,6,B
:PRINT" "+FL$(N);:LOCATE0,23:ATT
R2,5 ELSELOCATE15,2:ATTR3,6,U:PR
INT" ";:LOCATE0,23:ATTR2
,5
585 FORX=3TO22:LOCATE0,X:ATTR2,0
:PRINT" ";:PRINT" ";:LOCATE38,X:
ATTR2,0:PRINT" ";:NEXT:LOCATE0,2
3:ATTR2,5:PRINT" "
590 PALETTE5,31:HSCREEN0:GOSUB17
0
600 IFQ$="R"GOSUB950:GOSUB1100:G
OTO4000
610 IFQ$="F"ANDN GOSUB1000:IFN G
OTO500ELSECLS:GOTO3000ELSEIFQ$="
F"THENCLS:GOTO3000
620 IFQ$=CHR$(4)THENHSCREEN2:GOT
O1600
630 IFQ$="C"GOSUB950:GOSUB1100:I
=C:GOSUB1500:GOSUB3600:GOSUB1000
:GOTO500
640 IFQ$="A"GOSUB950:GOSUB1100:G
OSUB1500:GOTO5000
650 IFQ$="L"ANDN GOSUB1000:IFN G
OTO500ELSECLS:GOSUB1100:GOTO500E
LSEIFQ$="L"THENCLS:GOSUB1100:GOT
O500
660 IFQ$="P"GOSUB950:GOSUB1100:G
OSUB1550:GOTO4000
665 IFQ$="M"THENLB=1:GOSUB950:GO
SUB1100:GOSUB1550:GOTO9000
670 IFQ$<>CHR$(189)GOTO590

```

```

679 'End Prog
680 POKE65497,1:GOSUB300:POKE154
,112:POKE155,132
700 LOCATE8,3:ATTR3,7,B:PRINT" *
SESSION FINISHED * ";
750 ATTR2,5:LOCATE5,6:ATTR2,4
755 PRINT":(C)Copyright- Clive W
insall:":ATTR2,5:PRINT" ";:LOCA
TE11,19:ATTR5,4,U:PRINT"-COMPUTE
R NORMAL-";:ATTR2,5:PRINT" ";
760 CLEAR200:IFPEEK(32767) THENP
CLEAR(PEEK(32767))
765 PG=(PEEK(27)*256ORPEEK(28))-
(PEEK(25)*256ORPEEK(26))
770 LOCATE2,11:ATTR3,6:PRINT" Me
mory =";1536*PEEK(32767)+MEM+207
;:PRINT" Incl 200 Bytes CLEAR ";:
ATTR2,5:PRINT" ":LOCATE7,9:PRINT
"Program now =";PG;" bytes"
780 LOCATE2,13:ATTR3,6:PRINT" *
and";PEEK(32767);"Low-Res Graph
ic Page(s) * ";:ATTR2,5:PRINT" "
;
790 LOCATE9,16:ATTR2,1:PRINT" *"
;128;"K Coco No.3 *";:ATTR2,5:
PRINT" ";:LOCATE0,21
795 FORX=1TO8:SOUND200,1:NEXT:PA
LETTERGB:POKE65496,25:CLEAR200,3
2767:POKE65496,0:ATTR2,4:END
799 'Screen-Fmt/Mesg Blk
800 LOCATE7,23:ATTR2,4:PRINTSTRI
NG$(24,32);
802 LOCATE12,23:IF(PEEK(282))THE
NATTR7,4,B:PRINT"*";:ATTR7,4:PRI
NT" UPPER-Case";ELSEATTR5,4,B:P
RINT"*";:ATTR5,4:PRINT" Lower-Ca
se";804 RETURN
810 POKE65497,1:LOCATE0,4:ATTR2,
4:FORX=4TO21:LOCATE0,X:PRINT$:;
NEXT:IFY THENPOKE65496,0
814 LOCATE0,23:ATTR2,5:PRINT$:;
LOCATE7,23:ATTR2,4:PRINTLEFT$(S$,
24);
815 LOCATE0,22:ATTR7,4,U:PRINT$:
:LOCATE0,4:ATTR2,4:RETURN
820 FORX=4TO6:LOCATE0,X:ATTR2,4:
PRINT$:;NEXT:LOCATE17,4:ATTR1,4
,U:PRINT" Note ";:LOCATE0,7:ATT
R2,0,U:PRINT$:;LOCATE0,8:ATTR2,
4:RETURN
825 LOCATE35,2:ATTR2,1:PRINTUSIN
G"###";C+XX;ATTR2,4:PRINT" ";:R
ETURN
830 FORX=2TO21:LOCATE0,X:ATTR2,4
:PRINT$:;NEXT:LOCATE0,22:ATTR7,
4,U:PRINT$:;
831 LOCATE0,3:ATTR2,1,U:PRINT$:;
:LOCATE0,7:PRINT$:;LOCATE0,20:P
RINT$:;GOSUB814:RETURN
840 LOCATE2,2:ATTR3,0:PRINT$(PF
);:ATTR2,4:PRINT" ";:ATTR3,6:PRI
NT" "+FL$(N)+" ";:ATTR2,4:PRINT
";:LOCATE2,2
845 ATTR2,1:PRINTUSING"% %"
##";"Recd's";C;ATTR2,4:PRINT
" ";:LOCATE0,3:ATTR7,4,U:PRINT$:
;GOSUB814:LOCATE0,4:ATTR2,4:PRI
NT" ";:RETURN
855 LOCATE0,22:ATTR4,7,U:PRINT$:
:LOCATE0,23:ATTR2,5:PRINT" ";
:RETURN*'Red/Blk
860 LOCATE7,23:ATTR7,4:PRINT" To
END- Type In ";:ATTR2,4:PRINT"<
";:ATTR3,7,B:PRINT" FIN ";:ATTR2
,4:PRINT"> ";:RETURN
870 LOCATE2,6:ATTR5,4:PRINT"You
can Alter";:ATTR7,4:PRINT"U/";:

```

```

ATTR5,4:PRINT"L-Case";:ATTR2,4:
PRINT<SHIFT> + [0] ";:RETURN
879 'Col-Lines
880 ATTR2,0,U:PRINT$:RETURN'Dk-
Blue/Blk
881 ATTR2,1,U:PRINT$:;RETURN'G1
d/Blk
887 ATTR4,7,U:PRINT$:;RETURN'Re
d/Blk
889 'Type-in Entries
890 DUS=INKEYS:POKE282,255:P$(Z)
="" :ST=L1:L=0:IFZ=1GOSUB860
892 QS=INKEYS:IFQS=""GOTO892ELSE
SOUND200,1
894 IFQS<>CHR$(13)ANDQS<>CHR$(8)
THENP$(Z)=P$(Z)+QS:L=L+1:LOCATEL
1,L2:PRINTQS;
896 IFQS=CHR$(13)GOTO897ELSEIFQS
=CHR$(8)ANDL1=>1ANDL1>1THENL1=L1-
1:L=L-1:P$(Z)=LEFT$(P$(Z),L):LOC
ATEL1,L2:PRINT " ";:GOTO892ELSEIF
QS=CHR$(8)THENL=0:L1=ST:P$(Z)=""
:LOCATEL1,L2:GOTO892
897 IFQS=CHR$(13)THENRETURN
898 L1=L1+1:IFZ=1GOSUB800:ATTR2,
4:GOTO892ELSEGOSUB802:ATTR2,4:GO
TO892
899 ' Flags test- File in MEM
900 IFN>0ANDTT=1THENT=1
910 RETURN
950 TT=1:RETURN
999 'Clear MEM
1000 POKE65497,1:HSCREEN2:PALETT
E5,25:GOSUB300:SOUND200,2:GOSUB2
912
1010 LOCATE11,2:ATTR1,4:PRINT " *
Clear- Memory ";:ATTR2,4:PRINT "
":GOSUB881:LOCATE0,4:ATTR7,5:PR
INT " ":LOCATE6,5:PRINT"Do you wa
nt ";:ATTR3,6:PRINTFL$(N);:ATTR7
,5:PRINT " again?"
1015 LOCATE6,10:ATTR3,5:PRINT " Y
es/ No =[ Y/N ]. on Yes ";:LO
CATE9,12:PRINT"File is Kept in M
emory":LOCATE9,15:PRINT"[";:DUS=
"* SAVE RELOADING ":LOCATE30,15:
ATTR3,5:PRINT"]"
1016 LOCATE0,1:ATTR4,7:PRINT$:;
LOCATE0,20:ATTR2,1:PRINT$:;LOCA
TE0,21:ATTR3,4:PRINT$:;GOSUB855
1018 FORX=3TO19:LOCATE2,X:ATTR2,
1:PRINT " ";:LOCATE36,X:PRINT " ";
:NEXT:FORX=2TO20:LOCATE1,X:ATTR2
,4:PRINT " ";:LOCATE37,X:PRINT "
":NEXT:FORX=2TO21:LOCATE0,X:ATTR
4,7:PRINT " ";:LOCATE38,X:PRINT "
":NEXT:LOCATE19,18:ATTR2,5
1020 PALETTE5,0:HSCREEN0
1025 L=1:LOCATE10,15:ATTR3,7,U:F
ORX=1TOLEN(DUS):FORZ=1TO80:NEXTZ
:LOCATE11,15:PRINTMID$(DUS,X+1,(
LEN(DUS)-X+1));" ";:LOCATE28-X,
15:PRINTLEFT$(DUS,X);" ";:QS=INK
EYS:IFQS=""THENNEXT ELSEX=LEN(DU
S):IFQS=""GOTO1025
1030 IFQS="Y"GOTO1070
1035 IFQS<>"N"GOTO1020ELSECO=0:S
CHS="" :LB=0:TW=SW
1039 'Auto-Clr
1040 GOSUB1400:FOR I=1TOC+1
1050 FORZ=1TOF:A$(I,Z)="" :NEXTZ,
I:GOSUB1400:X=MEM:X=0
1060 PALETTE5,25:N=0:W=SW:TW=SW:
LB=0:TT=0:CO=0:C=0:RC=0:I=1:SCH$
="" :RETURN
1070 W=SW:TW=SW:LB=0:ZZ=0:ZX=0:C
C=0:RETURN

```

```

1099 'Load File
1100 T=0:I=1
1110 GOSUB900
1120 IFT THENRETURN
1130 POKE65497,1:HSCREEN2:PALETT
E5,56:GOSUB300:GOSUB2200:TT=0
1135 LOCATE3,21:PRINT"Enter req'
d ";:ATTR3,6:PRINT " INFO+F#'s ";
:ATTR2,1:PRINT " [#] ";:ATTR2,5:L
INEINPUT N$:N=VAL(N$):IFN<1THENC
LS:GOSUB1040:GOTO500ELSEIFN>K GO
SUB2000:GOTO1130
1136 PALETTE5,24:GOSUB300:LOCATE
5,9:ATTR3,5,U:PRINT"PLACE 'DATA-
Tape' in RECORDER ";:ATTR3,5:PRI
NT " "
1137 LOCATE2,11:PRINT" Rewind ta
pe ";G$:ATTR3,7,B:PRINT"<PLAY>"
:ATTR2,5:PRINT" Now!":GOSUB1830
1140 GOSUB300:LOCATE6,4:ATTR2,4:
PRINT " * Loading to Memory now *
":ATTR2,5:LOCATE14,7:ATTR3,6:P
RINT " ";FL$(N);" ";
1150 ATTR2,5:PRINT " ":GOSUB1400
1155 IFN>K GOSUB2000:GOTO1130
1158 POKE65496,0:OPEN"1",#D,FL$(
N)
1160 IF EOF(D)GOSUB1200:RETURN
1170 FORZ=1TOF:INPUT #D,A$(I,Z):
NEXT
1180 IFAS(I,1)=""ORAS(I,1)=""*GO
SUB1200:RETURN
1190 I=I+1:GOTO1160
1200 C=I-1:I=1:CLOSE #D:RETURN
1399 'Wait S/Rtn
1400 LOCATE7,23:ATTR3,7,B:PRINTV
T$:ATTR2,5:RETURN
1499 'Alter File-No.?
1500 PF=5:HSCREEN2:GOSUB300:GOSU
B840:GOSUB820
1510 LOCATE3,6:ATTR7,4:PRINT"or
Change File Num. to new [";:ATTR
2,1:PRINT " # ";:ATTR7,4:PRINT"]"
:ATTR2,4:PRINT " ";
1520 LOCATE3,5:ATTR2,4:PRINT " To
keep same Numb. ENTER [";:ATTR2
,1:PRINT " #";N::ATTR2,4:PRINT"]"
:HSCREEN0
1530 LOCATE0,10:ATTR2,5:PRINT:LO
CATE18,10:ATTR2,1:LINEINPUT " # "
:N$:ATTR2,5:N=VAL(N$):IFN=""GOT
O1530ELSEIFN<1ORN>K GOTO1530
1540 RETURN
1549 'Set-up Printer
1550 GOSUB300:PK=FNC(Y):IFPK GOS
UB2050:PK=FNC(Y):IFPK GOT0500
1555 Y=-2:GOSUB300:ATTR2,4:PRINT
" ";:ATTR3,6:PRINT " "+FL$(N)+"
":ATTR2,4:PRINTF$(8);:ATTR2,5:I
FLB THENPRINTF$(9):GOSUB887 ELSE
PRINTCHR$(13);:GOSUB887
1557 GOSUB855:POKE65496,0:IFLB T
HENW$="32":W=VAL(W$):GOSUB1575:G
OTO1565ELSEGOSUB1575
1559 'Codes-DMP105,CHR$(No)- 27+
19=normal:27+20=condensed:27+23=
Elite(semi-cond):27+14/15=stretc
h/end:27+31/32=bold/end:15/14=U-
Line/end
1560 LOCATE2,6:LINEINPUT"ENTER P
rinter-Width: ( 80/132 ).";W$:W
=VAL(W$):IFW<80ANDW<132THENLOC
ATE0,6:PRINTSTRING$(SW,32):GOTO1
560
1565 IFW=>132THENW=132:PRINT#-2,
CHR$(27)+CHR$(20) ELSEIFW<132THE
NW=80:PRINT#-2,CHR$(27)+CHR$(19)

```

```

1570 POKE155,W:POKE154,(W-16):PO
KE153,16:GOSUB855:ATTR2,5:RETURN
1575 LOCATE3,12:LINEINPUT"BAUD R
ATE [ 600/1200/2400] ";BD$:BD=VAL
(BD$):IFBD=0THENBD=600:BD$="600"
1580 IFBD=>2400THENBD=18 ELSEIFB
D=>1200THENBD=41 ELSEBD=88
1585 POKE150,BD:V=0:L=0:ST=0:TB=
0:RETURN
1599 'Info Pages 1-2
1600 POKE65497,0:PALETTE5,24:GOS
UB300:LOCATE1,2:ATTR3,0
1605 PRINT " *Infopage:1* ";:ATTR
2,4:PRINT " MEM(Free)="" :PRINTU
SING"####,% %";MEM;"Bytes":GO
SUB887:GOSUB820
1610 LOCATE2,5:PRINT"A possible"
;MR;" Records ea.":CR;" Char's";
1615 LOCATE3,6:ATTR5,4:PRINTK;"F
iles, each";FI;" Bytes total.":
LOCATE0,8:ATTR2,5:LOCATE8,8:ATTR
3,5,U:PRINT" ( Overhead-";:PRINTU
SING"####,% %";FM-MEM;" Byt
es ");:LOCATE0,10:ATTR2,5
1618 LOCATE10,10:PRINT"Program =
";PG;"Bytes."
1620 LOCATE4,12:ATTR5,4:PRINT " e
g:File #1(Business)=" :ATTR3,6:
PRINT " "+FL$(1);:ATTR2,5:PRINT
" "
1622 LOCATE0,19:GOSUB880:ATTR2,4
:PRINT " The Information - Colla
ting - Program":GOSUB880:ATTR2,5
1624 LOCATE3,16:IFN THENATTR3,7,
B,U:PRINT " File in Mem. ";:ATTR3,
6:PRINT " "+FL$(N);:ATTR2,1:PRINT
"=";C;"Records ";:ATTR2,5:PRINT
" " ELSELOCATE8,16:ATTR2,1,B:PRIN
T" No File in Memory now ";:ATTR
2,5:PRINT " "
1625 HSCREEN0:GOSUB1830:HSCREEN2
:PALETTE5,29:GOSUB300:GOSUB810:L
OCATE0,2:ATTR2,1,U:PRINT " * Elem
ents ";:ATTR3,7,U:PRINT"<<= Reco
rds =>>";:ATTR2,1,U:PRINT" Examp
les * "
1630 LOCATE0,4:ATTR2,4:L=4:FORX=
1TOF:L=L+2
1640 LOCATE1,L:ATTR7,4:PRINTUSIN
G"##";X;:ATTR5,4:PRINTNMS(X);:L
OCATE(39-LEN(EX$(X))),L:ATTR2,4:
PRINTEX$(X);
1645 NEXT:LOCATE11,4:ATTR5,4,U:P
RINT " * Infopage: 2 * ";:LOCATE0,
17:ATTR7,4,U:PRINT$:
1647 LOCATE0,18:ATTR5,4:PRINT"
Use 3 or MORE Char's to ";:ATTR1
,4:PRINT"Global-Search":LOCATE0,
19:ATTR5,4:PRINT" for Unknown
s When reading Files":LOCATE0,21
1648 FORX=2TO37:LOCATEX,21:ATTRR
ND(8-1),RND(8-1):PRINT"*";:NEXT:
LOCATE0,22:ATTR7,4,U:PRINT$: *A
ll F-B/Gnd Cols
1650 HSCREEN0:GOSUB1830:GOTO500
1799 'Flow/Scroll Ctl
1800 LOCATE7,23:ATTR7,4,B:PRINT"
Search- End ";:ATTR2,4:PRINT"Pr
ess:[";:ATTR3,7,B:PRINT" F1 ";
1820 ATTR2,4:PRINT"] ";:SOUND200
,4:GOTO1840ELSEGOTO1840
1830 LOCATE7,23:ATTR7,4:PRINT " T
o Continue ";:ATTR2,4:PRINT"Pres
s:[";:ATTR3,7,B:PRINT" F1 ";:ATT
R2,4:PRINT"] ";
1835 ATTR2,5:PRINT " ";:SOUND200,
2

```

```

1840 DUS=INKEY$
1850 Q$=INKEY$:IFQ$(<)CHRS(103)GO
TO1850
1855 IFPF=2THENLOCATE0,10:ATTR2,
5:PRINTSS$
1860 ATTR2,5:RETURN
1999 'Sub-Rtn Blks
2000 GOSUB300:IFD=-1ANDER<>20ORD
=1ANDER=20THENLOCATE4,10:ATTR3,7
,B:PRINT" FILE NON-EXISTANT CHE
CK [#] ! ";ATTR2,5:PRINT" "
2005 IFER=20THENLOCATE4,12:ATTR2
,4,B:PRINT" I/O Error- Check Tap
e/Disk!! ";ATTR2,5:PRINT" ":LOC
ATE9,16:ATTR2,4,U:PRINT" When ba
ck at- Menu: ";ATTR2,5
2010 CLOSE#D:ER=0:EL=0:GOSUB855:
ATTR2,5:GOSUB1830:RETURN
2050 LOCATE9,10:ATTR3,7,B:PRINT"
Printer NOT On-Line ! ";ATTR2,
5:PRINT" ":SOUND180,20:GOSUB1830
:RETURN
2199 'Files/Dir S/Rtns
2200 GOSUB830:LOCATE2,2:ATTR2,4:
PRINT" * Files available : ";A
TTR1,4:PRINT"for Selection..";:L
OCATE0,4:ATTR2,4
2210 LOCATE7,4:ATTR3,6:PRINT" "+
FL$(1)+" ";ATTR2,4:PRINT" TO "
;ATTR3,6:PRINT" "+FL$(K)+" ";:A
TTR2,4:PRINT" ";
2215 LOCATE2,8:ATTR5,4,U:PRINT"*
Disk-DIR space: ";:LOCATE0,5:AT
TR2,4
2220 LOCATE4,5:ATTR2,4:PRINT"Sel
ect a number between ";ATTR2,1:
PRINT" 1 & ";PRINTUSING"###";K;
:PRINT" ";ATTR2,4:PRINT" ";
2230 LOCATE4,14:ATTR1,4:PRINT" (
Try to keep Files in sequence,)"
;:LOCATE5,16:PRINT" ( when making
-up a File-Tape!)"
2240 LOCATE7,23:ATTR7,4:PRINT" T
o Abort, press-<";ATTR3,7,B:PRI
NT"ENTER";ATTR7,4:PRINT">";ATT
R2,4:PRINT" ";
2250 HSCREEN0:RETURN
2899 'Add New-Recd
2900 X=1:RN=0
2902 IFAS(X,1)<>"ANDAS(X,1)<>"*
"THENX=X+1:GOTO2902
2904 RN=X:X=0:RETURN
2909 'Set Func-Varb's
2910 SCH$="":I=1:CO=0:PT=0:XX=0
2912 IFC<RC THENC=RC:RETURN ELSE
RC=0:RETURN
2999 'New-File Funct
3000 PALETTE5,56:POKE65497,1:HSC
REEN2:GOSUB300:GOSUB2200
3030 LOCATE4,6:ATTR5,4:PRINT" C
heck for ";ATTR7,4:PRINT"1st Nu
mber";ATTR5,4:PRINT" available.
";
3040 LOCATE5,21:ATTR1,4:PRINT"En
ter the selected ";ATTR2,1:PRIN
T" Number.";:LINEINPUT N$:N=VAL(
N$):ATTR2,5
3045 IFN>K GOSUB2000:GOTO3000 EL
SEIFN<1THENCLS:GOSUB1040:GOTO500
3050 GOTO3500
3499 'New-Entry
3500 PF=3:PALETTE5,31:HSCREEN2:G
OSUB300:I=1:C=0:GOSUB840:GOSUB81
0:GOSUB820:GOSUB870
3510 LOCATE0,12:ATTR7,4,U:PRINTS
$;:HSCREEN0
3515 GOSUB3520:GOTO3580
3520 GOSUB820:GOSUB870:LOCATE14,
13:ATTR2,1:PRINT"Record No.";I;
3530 GOSUB860:LOCATE0,15:ATTR2,4
:FORZ=1TOF
3540 ATTR7,4:PRINT" ";Z;:ATTR5,
4:PRINTNM$(Z)+" ";:ATTR2,4
3550 HSTATDUS,A,L1,L2:GOSUB800:A
TTR2,4:GOSUB890
3555 IFP$(1)=""THENLOCATE0,15:GO
TO3550 ELSEIFI=1ANDP$(1)=""FIN"TH
ENZ=F:GOTO3570
3560 IFP$(1)=""FIN"ANDI=>1THENAS(
I,1)=""*:Z=F
3570 LOCATE0,L2+1:AS(I,Z)=P$(Z):
NEXT:RETURN
3580 IFA$(I,1)=""ORA$(I,1)=""*TH
ENI=I-1:RC=0:GOSUB3600:IFRC GOSU
B1000:GOTO500ELSEGOSUB1040:GOTO5
00
3590 IFI=>MR THENI=MR:GOSUB3600:
GOSUB1000:GOTO500
3595 GOSUB810:LOCATE0,8:GOSUB400
:XX=0:LOCATE36,2:ATTR2,1:PRINTUS
ING"###";I;ATTR2,4:PRINT" ";:I=
I+1:GOTO3515
3599 'Write File to Tape/Disk
3600 PF=2:HSCREEN2:PALETTE5,24:G
OSUB300:C=1:I=1:GOSUB840:GOSUB38
00
3605 GOSUB820:LOCATE0,10:ATTR2,5
:PRINTSS$;:LOCATE3,5:ATTR2,4:PRIN
T"Place DATA-Tape in Recorder NO
W !";:LOCATE2,6:ATTR5,4:PRINT"Ig
nore ";ATTR7,4:PRINT"<PLAY> ";
3606 ATTR5,4:PRINT"to CSAVE @ Cu
rr't-Pos!";ATTR2,4:PRINT" ";:GO
SUB3710
3607 GOSUB3722:GOSUB820:LOCATE2,
6:ATTR5,4:PRINT"You may ignore "
;ATTR7,4:PRINT"Recorder";ATTR5
,4:PRINT" if you wish!";:LOCATE0
,10:ATTR2,5:PRINTSS$
3608 LOCATE5,10:PRINT" Now Press
: ";ATTR3,7,B:PRINT"<RECORD>";:
ATTR2,4,B:PRINT"<PLAY>";ATTR2,
5:PRINT" Keys";
3610 GOSUB1830:LOCATE10,5:ATTR2,
4:PRINT"SAVE(ing).";ATTR3,6:PRI
NT" "+FL$(N)+" ";ATTR2,4:PRINT"
":LOCATE0,8:GOSUB1400
3615 IFC<1THENRC=0:I=1:RETURN EL
SEIFC>MR THENC=MR
3620 POKE65496,0:OPEN"O",#D,FL$(
N)
3630 FORI=1TOC
3640 IFA$(I,1)=""ORA$(I,1)=""*TH
ENI=C:GOTO3670
3650 FORZ=1TOF:PRINT #D,AS(I,Z):
NEXT:RC=RC+1
3670 NEXT
3680 CLOSE#D:C=0:LOCATE0,10:PRIN
TSS$;
3690 Z=TL:TL=TL/5:GOSUB3700:GOSU
B3722:TL=Z:Z=0:RETURN
3699 'Tape-Recd Rtns
3700 GOSUB820:LOCATE7,5:ATTR2,4:
PRINT"Now Release ";ATTR3,7:PRI
NT"<RECORD>";ATTR2,4:PRINT" Key
":LOCATE3,6:PRINT"while holding
";ATTR0,4:PRINT"<STOP>";ATTR2,
4:PRINT" / ";ATTR5,4:PRINT"<PAU
SE>";ATTR2,4:PRINT" Key";
3710 LOCATE0,10:ATTR2,5:LOCATE5,
10:ATTR2,5:PRINT" "+G$;ATTR2,4:
PRINT"<";ATTR2,4,B:PRINT"PLAY";
ATTR2,4:PRINT">";ATTR2,5:PRINT
" Key ONLY!"
3720 GOSUB855:GOSUB1830:RETURN
3722 GOSUB820:LOCATE2,6
3725 ATTR2,4:PRINT"* ";ATTR7,4,
B:PRINT"Spacing ";ATTR2,4:PRINT
": of Files now occurring! ";:LO
CATE0,10:ATTR2,5
3730 GOSUB1400:POKE65496,0:MOTOR
ON:FORM=1TOTL:NEXT:MOTOROFF
3740 POKE65497,1:LOCATE0,10:ATTR
2,5:PRINT:RETURN
3799 'SKIPF S/Rtn
3800 AUDICON:GOSUB820
3810 LOCATE3,5:ATTR2,4:PRINT"SAV
E @ Present Tape-Position";ATT
R5,4,B:PRINT"<ENTER>";ATTR2,4:P
RINT" ";
3815 LOCATE2,6:PRINT"Csave past
";ATTR3,6:PRINT" Last-File ";:A
TTR2,4:PRINT"=Enter ";ATTR2,1:P
RINT"Last #!";ATTR2,4:PRINT" ";
:LOCATE0,10:ATTR2,5:LOCATE0,10:A
TTR2,5:PRINTSS$:LOCATE18,10
3820 HSCREEN0:ATTR2,1:LINEINPUT"
";LNS:IFLNS=""GOTO3840ELSELN=VAL
(LNS):IFLN=>1ANDLN=<K GOSUB820:G
OSUB3710ELSEGOTO3815
3830 IFLN GOSUB820:LOCATE1,5:ATT
R2,4:PRINT"Now SKIPF(orward) Pas
t ";ATTR3,6:PRINT" "+FL$(LN)+"
";ATTR2,4:PRINT" ":POKE65496,0:
SKIPF FL$(LN):POKE65497,1
3840 GOSUB1400:RETURN
3999 'Read/Search Multi-use Func
4000 PF=4:GOSUB2910
4005 HSCREEN2:IFY ANDLB=0THENPAL
ETTE5,31:TW=W:W=79:GOSUB300:W=TW
:TW=SW ELSEPALETTE5,24:GOSUB300
4010 HSCREEN2:GOSUB840:GOSUB820:
POKE65496,0
4020 LOCATE2,5:ATTR5,4,U:PRINTCE
$;"Information Required!";ATTR
2,4:PRINT" ";
4030 GOSUB860:LOCATE0,6:ATTR2,4
4040 LOCATE2,6:PRINTCES+["";ATT
R5,4:PRINTSCH$;ATTR2,4:PRINT"]"
;:LOCATE0,8:ATTR2,5:PRINT" ":IFY
THENDV=Y ELSEDV=0
4050 LOCATE3,16:ATTR2,1:PRINT" "
;COUNT;" Records Searched";ATTR
2,1,B:PRINTCC;ATTR2,1:PRINT" Re
sult(s)";ATTR2,5:PRINT" ";:LOCA
TE0,8
4055 IFY THENATTR2,4:PRINT" "+F
$(8)+"WIDTH";:PRINTUSING"###";W;
:PRINT" Baud=";BD$;ATTR2,5:PRIN
T" "
4057 IFLB THENLOCATE13,19:ATTR2,
4,U:PRINTF$(9);ATTR2,5:PRINT" "
;
4060 HSCREEN0:LOCATE3,10:PRINT"
Enter Known ( needed ) - Details
";:LOCATE4,12:LINEINPUT SCH$
4065 PT=0:CC=0:CO=0:I=1:LOCATE0,
11:GOSUB880:LOCATE0,12:ATTR2,5
4067 IFLB ANDSCH$=""FIN"GOTO9000
4070 IFSCH$=""FIN"GOSUB290:DV=0:X
X=0:Y=0:W=SW:RC=C:GOSUB1000:GOTO
500
4080 IFDV=0THENPOKE65497,1:GOSUB
300ELSEPOKE65496,0
4090 ZZ=0:P=0
4095 FORZ=1TOF
4100 IF LEN(SCH$)>2THENP=INSTR(1
,AS(I,Z),SCH$)
4105 IFP THENZ=F:ZZ=1:GOTO4112
4110 IFA$(I,Z)=SCH$THENZ=F:ZZ=1
4112 NEXT

```

4115 IFAS(I,1)="ORAS(I,1)="*GO
SUB5350:GOSUB1800:GOSUB490:GOTO4
010
4125 CO=CO+1
4127 IFLB ANDZZ GOSUB460:GOTO414
0
4130 IFZZ=1GOSUB330
4140 I=I+1:GOTO4090
4999 'Alter-File
5000 PF=5:POKE65497,1:HSCREEN2:P
ALETTE5,31:GOSUB300:XX=0
5010 GOSUB840:GOSUB825:GOSUB810:
GOSUB820:LOCATE2,5:ATTR5,4:PRINT
"You may Add/Edit old/new Record
s,":LOCATE3,6:PRINT"staying as
long as you wish to!";LOCATE0,1
0:ATTR2,5:HSCREEN0
5015 IFRN>MR THENATTR3,7,B:PRINT
" FILE FULL! NO MORE RECORDS NOW
";ATTR2,5:PRINT " ":GOSUB881:AT
TR2,5:GOTO5880
5020 GOSUB150
5040 IFQ\$="E"ANDXX=0ANDZX=0THENG
OSUB1000:GOTO500
5050 IFQ\$="E"THENRN=0:GOTO6000EL
SEIFQ\$="R"GOSUB2900:GOTO5800
5070 IFQ\$="C"THEN5200ELSE5020
5199 'Find Old-Rec
5200 PF=5:HSCREEN2:GOSUB300:I=1:
GOSUB840:GOSUB825:GOSUB810
5210 GOSUB820:LOCATE0,6:ATTR2,4:
LOCATE2,6:PRINT"Search for Old R
ec'd (" :ATTR5,4,B:PRINT"Be Spec
ific!";ATTR2,4:PRINT")";
5220 LOCATE0,12:ATTR2,4:HSCREEN0
5230 LOCATE4,12:PRINT"Enter 'Kno
wn'-detail.";
5240 LINEINPUT SCH\$:IFSCH\$="FIN"
THENHSCREEN2:GOTO5010
5250 ZZ=0:RN=0
5260 FORZ=1TOF
5270 IFAS(I,Z)=SCH\$THENZ=F:ZZ=1:
PT=1
5275 NEXT:IF AS(I,1)="OR AS(I,1
)="*GOSUB5350:GOTO5200ELSEIFZZ=
1GOTO5295
5290 I=I+1:GOTO5250
5295 RN=1:I=1:PT=0:GOTO5500
5349 'Valid-search?
5350 IFPT GOTO5380
5360 LOCATE0,12:ATTR2,5:PRINTS\$;
:LOCATE4,12:ATTR3,7,B:PRINT" NO
ELEMENT.[";SCH\$;"] HERE! ";ATTR
2,5:PRINT " "
5370 IFY THENLOCATE0,10:PRINTS\$
5380 RETURN
5499 'Alter-Options
5500 PF=5:HSCREEN2:GOSUB300:GOSU
B840:GOSUB825:GOSUB810:GOSUB820
5550 LOCATE2,5:ATTR5,4:PRINT"To
Completely Re-Do this Record: [C
J]"
5560 ATTR5,4:PRINT" or Alter a
n Element only Press: [A]";ATTR
2,4:PRINT " ";
5570 LOCATE5,21:ATTR7,4:PRINT"To
End Function Right now.[";ATTR
3,7,B:PRINT" E ";ATTR7,4:PRINT"
J";ATTR2,4:PRINT " ";
5590 HSCREEN0:LOCATE0,8:ATTR2,4:
I=RN:GOSUB400:XX=XX-1:LOCATE0,14
:ATTR2,4
5610 GOSUB170:IFQ\$="E"THENHSCREE
N2:GOTO5010ELSEIFQ\$="A"GOTO5640
5620 IFQ\$="C"THENXX=XX-1:5800ELS
E5590
5639 'Alter-Elmt

5640 GOSUB810:GOSUB820:GOSUB802
5650 LOCATE0,8:FORX=1TOF:ATTR7,4
:PRINT" ";X;"=" :ATTR5,4:PRINTN
M\$(X);ATTR2,4:PRINT" .";AS(RN,X
):NEXT:LOCATE0,8+F:ATTR2,4:HSTAT
DUS,A,L1,L2:GOSUB800:LOCATEL1+2,
L2+1:ATTR2,4
5660 LOCATE0,L2:ATTR7,4,U:PRINTS
\$;LOCATE0,L2+1:ATTR2,4
5680 LOCATE2,6:ATTR7,4:PRINT"Sel
ect Element-No. and enter it.":
ATTR2,4:PRINT " ":LOCATEL1+2,L2+
2:PRINT" Enter- ";ATTR7,4:LINEI
NPUT"";QS=X=VAL(QS):IFX<1ORX>F G
OTO5680
5700 ATTR5,4:PRINT" ";NM\$(X)+
";ATTR2,4:LINEINPUT P\$(X):ATTR2
,4
5710 AS(RN,X)=P\$(X):X=0:ZX=ZX+1
5720 LOCATE6,21:ATTR7,4:PRINT" N
ext-Element change =[N] ";ATT
R2,4:PRINT " ";
5730 GOSUB150:ATTR2,4:IFQ\$="N"TH
EN5640ELSE5900
5799 'Add New-Rec
5800 PF=5:HSCREEN2:GOSUB300:GOSU
B840:GOSUB825:GOSUB810:GOSUB820:
GOSUB870
5810 LOCATE0,12:ATTR7,4,U:PRINTS
\$;HSCREEN0:LOCATE0,14:ATTR2,4
5820 XX=XX+1:HSTATDUS,A,L1,L2:I=
RN:GOSUB3520
5840 IFAS(RN,1)="ORAS(RN,1)="*
THENXX=XX-1:HSCREEN2:GOTO5010
5860 RN=0:I=1:GOSUB825
5880 GOSUB150
5900 IFQ\$="C"GOTO5200ELSEIFQ\$="R
"GOSUB2900:GOTO5800
5910 IFQ\$="E"THENHSCREEN2:GOTO50
10ELSE5880
5999 'File altered Test
6000 PALETTE5,56:GOSUB300:LOCATE
1,2:ATTR7,4:PRINT" *Check if Fil
e Changed-";
6010 ATTR2,4:PRINT " ":LOCATE28,2
:ATTR3,6:PRINT" "+FL\$(N);ATTR2,
4:PRINT " ":GOSUB881:ATTR3,5
6030 IFXX=0ORZX=0GOTO6070
6040 XX=0:ZX=0:GOSUB1000:GOTO500
6070 LOCATE6,8:PRINT"You entered
";XX;" New-Record(s)"
6080 LOCATE4,12:PRINT" &";ZX;" N
ew-Detail(s) to ";FL\$(N)
6090 I=C+XX:C=0:LOCATE0,22:GOSUB
881:ATTR2,5:LOCATE0,20
6100 GOSUB1830:GOSUB3600:GOSUB10
00:GOTO500
7009 'DSK to DSK File-Copy Blk
8009 'DSK Multi-File Access Blk
8999 'Mail-Labels
9000 PF=9:POKE65497,1:PALETTE5,3
1:GOSUB490:GOSUB2910
9010 GOSUB840:GOSUB820:LOCATE10,
12:PRINT":WIDTH=";W\$;" BAUD=";BD
\$;LOCATE0,5
9020 LOCATE2,5:ATTR2,4
9030 PRINTMS\$;GOSUB855:ATTR7,4:
LOCATE4,23:PRINT"To Alter Messag
e-[A]";ATTR2,4:PRINT" Not=<ENT>
";
9040 LOCATE0,8:ATTR2,5:GOSUB170:
IFQ\$<"A"GOTO9060
9050 LOCATE2,5:ATTR7,4:MS\$="":PR
INT"60 Char-max. Message! (Erase =
<ENTER>)";ATTR2,4:PRINT " ";LO
CATE0,8:ATTR2,4:LINEINPUT " ";MS
\$:MS\$=LEFT\$(MS\$,60):ATTR2,5

9060 GOSUB820:LOCATE0,5:ATTR2,4:
PRINTMS\$;LOCATE0,21:ATTR2,5:LOC
ATE4,21:ATTR7,4:PRINT"If another
File is wanted - Yes!";ATTR2,5
:PRINT " ":GOSUB814
9070 LOCATE8,12:ATTR2,4,B,U:PRIN
T" Set Ribbon at Label-Top ";AT
TR2,5:PRINT " ":LOCATE1,23:ATTR7,
4:PRINT" To Print-Some/All/";AT
TR3,7:PRINT"End";ATTR7,4:PRINT"
-[";ATTR3,7,B:PRINT" E ";ATTR7
,4:PRINT"/A/S/Y) ";ATTR2,5
9080 GOSUB855:GOSUB170:POKE65496
,0
9090 IFQ\$="E"GOTO9220
9100 IFQ\$="S"THENDV=0:GOTO4000 E
LSEIFQ\$="Y"GOTO9200 ELSEIFQ\$<"A
"GOTO9080
9120 SCH\$=" All- Records! ":GOSU
B855:ATTR2,5:I=1
9130 PK=FNC(Y):IFPK GOSUB2050'*T
oggle"Paper-Switch"NOW!
9140 PK=FNC(Y):IFPK GOTO500
9150 GOSUB1400:GOSUB460:I=I+1:XX
=XX+1:IFI>C ORI>MR GOTO9000
9180 IFAS(I,1)="ORAS(I,1)="*TH
EN9000ELSE9130
9200 CO=0:SCH\$="":W=SW:GOSUB1040
:GOSUB1100
9210 LB=1:GOTO9000
9220 Y=0:LB=0:DV=0:W=SW:RC=0:MS\$
="":GOSUB1000:GOTO500
9999 'Error Traps(ADD MORE?)
10000 ER=ERNO:EL=ERLIN:PRINT#DV,
CHR\$(13)*Funct's dynamic, Varb'
s HOLD values!
10001 HSCREEN0:PRINTCHR\$(13):IFE
R=2ANDEL<1THENPRINTCHR\$(13);"I d
on't understand you?(Key/Brd Err
or)":GOTO10090
10002 IFER=1THENPRINT"I found a
Typing error in Line #";EL:GOTO1
0090
10003 IFER=7THENPRINT" You have
DEL(eted) /Lost LINE #";EL;CHR\$(
13);" Type 'GOTO500' & End, th
en Fix Prog!":GOTO10090
10010 IFER=20GOSUB2000:CLOSE#D:G
OSUB1040:GOTO500
10080 ATTR2,4,B,U:PRINT" Error
#";ER;" in line #";EL;" ";ATTR2
,5:PRINT " "
10090 POKE65496,0:ATTR2,4:SOUND1
80,20:STOP

Data storage is what this program is all about.

DATABASE

32K ECB CoCo + "Screen 51"
BUSINESS

by William Boardman

HEREWITH IS A modified version of Sam Robinson's "A Tape DataBase" designed to work with Chris Bone's "51-Column Screen" program (see April '85 Australian Rainbow). The 'PRINT @' statements will cause FC errors without it. Have fun!

The Listing:

```
0 GOTO15
1 '***** DBASE 51 *****
2 '***** WILLIAM BOARDMAN *****
3 SAVE"166:3":END
15 '*****A DATABASE
*****BY S
AM ROBINSON MODIFIED BY W.BOARDMA
N.*****FOR 16K DISK
SYSTEM AND SCREEN51*****
*****PROGRAM 14/8/85*****
*****
16 CLS:SCREEN1,1
17 PRINT@10,"TRS-80 USER GROUP O
F PT.LINCLON.";
18 PRINT@110,"ALWAYS ENTER DATE
(**/**/)**":PRINT@160,"AT BEGINN
ING OF EVERY COMMENT.":PRINT@213
,"ALWAYS END WITH INITIALS.":PRIN
T@266,"(SER) ETC."
19 PRINT@468,"PRESS <ENTER> TO C
ONTINUE ";
20 IFINKEY$=""THEN20
21 CLEAR3500:DIM S$(20)
22 'COMMENTS FOLLOW
23 CLS0
24 PRINT@10,"COMPUNUTS USER GROU
P .....";
25 PRINT@87,"DO YOU WANT TO--";
26 PRINT@115,"(1) START A NEW FI
LE"
27 PRINT@166,"(2) REPLACE A COMM
ENT"
28 PRINT@217,"(3) ADD TO EXISTIN
G FILE"
29 PRINT@268,"(4) DELETE A COMME
NT"
30 PRINT@319,"(5) PRINT COMMENTS
ON SCREEN"
31 PRINT@370,"(6) PRINT COMMENTS
TO PRINTER";
32 PRINT@421,"(7) SAVE COMMENTS
TO DISK"
33 PRINT@472,"(8) LOAD COMMENTS
FROM DISK"
34 PRINT@725,"*PLEASE MAKE YOUR
SELECTION";
36 PRINT@776,"(1-9)";
```

```
37 L$=CHR$(13)
38 A$=INKEY$:IF A$=""THEN38
39 M=VAL(A$)
40 IF M<10R M>9THEN38
41 CLS
42 ON M GOSUB 46,59,48,70,83,136
,113,124
44 GOTO22
45 '
46 'INPUT/ADD COMMENTS
47 Y=1
48 CLS:PRINT@17,"INPUT/ADD COMME
NTS"
49 PRINT@114,"PRESS <ENTER> WHEN
FINISHED"
50 PRINT:PRINT"DATE(**/**/)**";:
INPUT D$(Y)
51 PRINT"COMMENT"Y:PRINT"***MUST
BE SHORTER THAN 8 LINES**"
52 GOSUB140
53 PRINT:PRINT"INITIALS (***)";:
INPUT I$(Y)
54 CLS
55 IFD$(Y)=""THEN RETURN
56 Y=Y+1
57 GOTO50
58 '
59 'REPLACE COMMENTS
60 N=0
61 CLS:PRINT@18,"REPLACE COMMENT
S"
62 PRINT@114,"PRESS <ENTER> WHEN
FINISHED"
63 PRINT:INPUT"COMMENT NO. TO RE
PLACE";N
64 IF N=0THEN RETURN ELSE 65
65 PRINT"REPLACEMENT COMMENT.":P
RINT"***MUST BE SHORTER THAN 8 LI
NES**":GOSUB149
66 PRINT
67 GOTO59
68 RETURN
69 '
70 'DELETE COMMENTS
71 CLS:PRINT@16,"DELETE COMMENT"
72 PRINT@114,"PRESS <ENTER> WHEN
FINISHED"
73 PRINT:INPUT"COMMENT TO DELETE
";N
74 IFN=<0THEN RETURN
75 IFN>Y-1THEN 73
76 FOR X=N TO Y-2
77 D$(X)=D$(X+1):S$(X)=S$(X+1):I
$(X)=I$(X+1)
78 NEXTX
79 D$(X)="" : S$(X)="" : I$(X)=""
80 Y=Y+1
81 GOTO70
82 '
83 'PRINT COMMENTS
84 IF M=5THEN P=0
85 IFM=6THEN P=2
```

```
86 CLS0:PRINT@158,"WOULD YOU LIK
E TO PRINT -";:PRINT@259,"(A)LL
COMMENTS OR (O)NE COMMENT?";
87 A$=INKEY$:IF A$=""THEN87
88 IF A$<>"A"AND A$<>"O"THEN87
89 IF A$="A"THEN93
90 IF A$="O"THENPRINT@352,"PRINT
WHICH COMMENT?";
91 B$=INKEY$:IF B$=""THEN91
92 X=VAL(B$)
93 CLS:IF P=2THEN94ELSE96
94 CLS0:PRINT@236,"PRINTING NOW"
;
95 IFP=2THENGOSUB137
96 IF A$="O"THEN98
97 FORX=1TO Y-1 STEP 1
98 IFP=0THENGOSUB137
99 PRINT#-P,X". ";D$(X);L$;S$(
X);L$;I$(X);L$;STRING$(32,"-")
100 IFP=2THEN GOTO104ELSE IFP=0T
HEN GOTO101
101 PRINT" PRESS <ENTER> TO CON
TINUE"
102 C$=INKEY$:IF C$=""THEN102
103 CLS
104 IF A$="O"THENGOTO23
105 NEXTX
106 RETURN
107 IF EOF(-1)THEN111
108 PRINT#-P,D$(Y);S$(Y);I$(Y);
109 Y=Y+1
110 GOTO107
111 CLOSE#-1:RETURN
112 '
113 'SAVE COMMENTS ON DISK
114 CLS:PRINT@113,"SAVE COMMENTS
ON DISK"
115 PRINT@221,"READY DISK"
116 PRINT@418,"PRESS <ENTER> WHE
N READY"
117 A$=INKEY$:IF A$=""THEN117
118 OPEN"O",#1,"COMMENTS"
119 FORX=1TO Y-1
120 WRITE#1,D$(X),S$(X),I$(X)
121 NEXTX
122 CLOSE#1:RETURN
123 '
124 'LOAD COMMENTS FROM DISK
125 CLS:PRINT@113,"LOAD COMMENTS
FROM DISK"
126 PRINT@221,"READY DISK"
127 PRINT@418,"PRESS <ENTER> WHE
N READY"
128 R$=INKEY$:IF R$=""THEN128
129 OPEN"1",#1,"COMMENTS"
130 Y=1
131 IF EOF(1)THEN135
132 INPUT#1,D$(Y),S$(Y),I$(Y)
133 Y=Y+1
134 GOTO131
135 CLOSE#1:RETURN
136 GOTO83
```

```

137 PRINT#-P,"COMPUNITS USER GRO
UP ....."
138 PRINT#-P,"-----"
-----"
139 RETURN
140 AS=INKEY$
141 IF AS="" THEN PRINT CHR$(191
);CHR$(8);:GOTO140
142 IF AS=CHR$(13) THEN BS="":RE
TURN
143 IF POS(P) >22 THEN IF AS=CHR$(
32) THEN AS=CHR$(13)
144 PRINTAS;
145 BS=BS+AS
146 SS(Y)=BS
147 GOTO140
148 RETURN
149 CS=INKEY$
150 IF CS="" THEN PRINT CHR$(191
);CHR$(8);:GOTO149
151 IF CS=CHR$(13) THEN DS="":RE
TURN
152 IF POS(P) >22 THEN IF CS=CHR
$(32) THEN CS=CHR$(13)
153 PRINTCS;
154 DS=DS+CS
155 SS(N)=DS
156 GOTO149

```

More.....

MUSIC

32K ECB + "MUSIC+"

by D. Voutsis

I HAVE FOUR extra music pieces for your music competition. My daughter (13) has written the first three, (Music Box) DANCER, BARCARO(11e) and FOLK (Dance), while I composed an old favourite by Roy Orbison, WMN (Pretty Woman).

They have all been put together using MUSIC+. In other words the ML code for these pieces is on the end of this month's CoCoOz tape, and must be used in conjunction with this program.

Enjoy!

(Ed's Note: Due to the length of the last two pieces, we are unable to present them here in the magazine, but instead will appear at the end of this month's CoCoOz.)

34: 4. ,B4 ,B4 ,B3 ,G3

FOLK

COL:	LEN	,V1	,V2	,V3	,V4
1:	8	,B3	,B3	,0	,0
2:	8	,C4	,C4	,0	,0
3:	4	,D4	,D4	,G2	,0
4:	4	,B3	,B3	,D3	,0
5:	4	,B3	,B3	,D3	,0
6:	4	,B3	,B3	,C3	,0
7:	4	,A3	,A3	,D3	,0
8:	8	,A3	,A3	,D3	,0
9:	8	,B3	,B3	,D3	,0
10:	4	,C4	,C4	,A2	,0
11:	4	,A3	,A3	,D3	,0
12:	4	,A3	,A3	,D3	,0
13:	4	,D4	,D4	,B2	,0
14:	4	,D4	,D4	,D3	,0
15:	8	,B3	,B3	,0	,0
16:	8	,C4	,C4	,0	,0
17:	4	,D4	,D4	,G2	,0
18:	4	,B3	,B3	,D3	,0
19:	4	,B3	,B3	,D3	,0
20:	4	,B3	,B3	,C3	,0
21:	4	,A3	,A3	,D3	,0
22:	8	,A3	,A3	,D3	,0
23:	8	,B3	,B3	,D3	,0
24:	4	,C4	,C4	,A2	,0
25:	4	,A3	,A3	,D3	,0
26:	4	,A3	,A3	,D3	,0
27:	2	,G3	,G3	,D3	,0
28:	8	,A3	,A3	,C3	,0
29:	8	,B3	,B3	,C3	,0
30:	4	,C4	,C4	,D3	,0
31:	4	,C4	,C4	,D3	,0
32:	8	,B3	,B3	,G2	,0
33:	8	,A3	,A3	,G2	,0
34:	4	,G3	,G3	,D3	,0
35:	4	,G3	,G3	,D3	,0
36:	8	,A3	,A3	,C3	,0
37:	8	,B3	,B3	,C3	,0
38:	4	,C4	,C4	,D3	,0
39:	4	,C4	,C4	,D3	,0
40:	4	,B3	,B3	,G2	,0
41:	4	,B3	,B3	,D2	,0
42:	4	,B3	,B3	,D2	,0
43:	8	,A3	,A3	,C3	,0
44:	8	,B3	,B3	,C3	,0
45:	4	,C4	,C4	,D3	,0
46:	4	,C4	,C4	,D3	,0
47:	8	,B3	,B3	,G2	,0
48:	8	,A3	,A3	,G2	,0
49:	4	,G3	,G3	,D3	,0
50:	4	,G3	,G3	,D3	,0
51:	4	,A3	,A3	,C3	,0
52:	4	,B3	,B3	,D3	,0
53:	4	,A3	,A3	,D3	,0
54:	2	,G3	,G3	,B2	,G2

BARCARO

COL:	LEN	,V1	,V2	,V3	,V4
1:	4	,B4	,B4	,G3	,0
2:	8	,C5	,C5	,G3	,0
3:	4	,C5	,C5	,D4	,0
4:	8	,B4	,B4	,D4	,0
5:	8	,B4	,B4	,C4	,0
6:	8	,A4	,A4	,C4	,0
7:	8	,C5	,C5	,C4	,0
8:	4	,C5	,C5	,D4	,0
9:	8	,B4	,B4	,D4	,0
10:	8	,B4	,B4	,C4	,0
11:	8	,A4	,A4	,C4	,0
12:	8	,C5	,C5	,C4	,0
13:	4	,C5	,C5	,D4	,0
14:	8	,B4	,B4	,D4	,0
15:	4	,B4	,B4	,G3	,0
16:	4	,B4	,B4	,D4	,0
17:	4	,B4	,B4	,G3	,0
18:	8	,C5	,C5	,G3	,0
19:	4	,C5	,C5	,D4	,0
20:	8	,B4	,B4	,D4	,0
21:	8	,B4	,B4	,C4	,0
22:	8	,A4	,A4	,C4	,0
23:	8	,C5	,C5	,C4	,0
24:	4	,C5	,C5	,D4	,0
25:	8	,B4	,B4	,D4	,0
26:	8	,B4	,B4	,C4	,0
27:	8	,A4	,A4	,C4	,0
28:	8	,C5	,C5	,C4	,0
29:	4	,C5	,C5	,D4	,0
30:	8	,B4	,B4	,D4	,0
31:	8	,B4	,B4	,G3	,0
32:	8	,B4	,B4	,B3	,0
33:	8	,B4	,B4	,D4	,0

HINT.....

Printer baud rates

I have found a POKE for the Colour Computer that may be of interest to many users. When running the computer with a high speed POKE, you can print out to a printer if you POKE location 150 with your printer baud rate at multiply this figure by two.

For example:-

-POKE65495,0:POKE150,88*2

High speed CoCo with printer running at 600 baud.

- POKE65495,0:POKE150,41*2

High speed CoCo with printer running at 1200 baud.

You will be able to print to the printer without returning the computer to normal speed. I hope this information will be of use to you, because I haven't read about it anywhere in my magazines.

CASHBOOK

OS-9
BUSINESS

by Ian Lobley

(The following program is in Basic09 source code and is quite long. Several Procedures will be published each month and will then be followed by Ian's articles explaining the operation of the program. As Ian uses the HIRES screen, those who wish to use Wordpak will have to alter the Modules accordingly.

This program won Ian the 68008 Class Computer (CoCo add on) donated by Blaxland Computers at the recent CoCo Conf. Well done Ian. Ed.)

I started the program a while ago mainly to learn OS9 and BASIC09. The main attraction of the program is the ability to be able to enter codes (instead of numbers) when posting cheques. Also you can insert codes and categories during a posting session. The yearly printout calculates and prints out the yearly expenses into monthly totals and category totals. Anyone who runs it should be able to, at least halve his accountant's fee. If he doesn't, he needs a new accountant.

The program is written in BASIC09 with most of the procedures put into the BOOT file. There are four files generated by the program, these are called CHQ_RECORD COMPANY_NAME BANK and ITEMS. The first file holds the Date, cheque No., payee, the amount and the Code. The ITEMS list file contains the code and category of the item.

The program will allow the user to enter 30 different expense categories, each with their own particular code that is easily remembered when entering cheques. For example, a code such as CARS would mean Car Service Expenses, or TELE for telephone expenses. This is easier than using numbers such as 1001 1002 for the codes. These codes do not have to be entered at the start of the program, but can be entered during a cheque posting session.

It will allow the user to

enter deposit items as well as printout a monthly itemised list of individual cheques, expenses and deposits. At any time a yearly printout of all data can be obtained.

```
RUN CASH includes the
following procedures :-
menu Read_Bank Title
title2 items Change_chq
get_chq search_chq
create_chq print_t
print_chq set_date
input_chq plus
```

```
CREATE_CASH includes the
following procedures :-
create_menu create_chq
create_bank create_name
input_items create_items
```

```
*REM create_bank creates
bank and start_bal
```

```
PRINT_CASH
search_month_p print_chq_p
title_p macau_title items
```

```
OTHERS
Search_month
delete_chq_record del_all
see_bank recon
recalc change_start_bal
```

****PLEASE NOTE****
that lines indented by two
(2) characters are in fact a
continuation of the previous
line.

```
PROCEDURE create_menu
DIM choice:STRING[1]
PRINT CHR$(12)
RUN create_items
RUN input_items
RUN create_name
RUN create_bank
```

```
RUN create_chq
END
PROCEDURE create_chq
PRINT CHR$(12)
TYPE record=date:STRING[8];
      number:INTEGER; payee:
      STRING[25]; ammt:REAL;
      item:STRING[4]
DIM chq_rec:record
DIM path:BYTE
chq_rec.date=""
chq_rec.number=0
chq_rec.payee=""
chq_rec.ammt=0
chq_rec.item=""
CREATE #path,"chq_record"
PUT #path,chq_rec
CLOSE #path
PROCEDURE create_bank
DIM choice:STRING[1]
DIM balance:REAL
PRINT CHR$(12)
DIM path:BYTE
balance=0
PRINT "Next we have to set
      up your bank Account "
PRINT
PRINT "Input your beginning
      of the Financial Year Bank
      Balance"
INPUT " > ",balance
CREATE #path,"bank"
PUT #path,balance
CLOSE #path
CREATE #path,"start_bal"
PUT #path,balance
CLOSE #path
END
PROCEDURE create_name
PRINT CHR$(12)
DIM name:STRING[20]
DIM path:BYTE
PRINT "The next thing we
      have to do is to enter the
      name you want"
PRINT "used when you
      generate any Printouts "
PRINT
PRINT "You have space for up
      to 20 Characters "
INPUT "Type Name in Here >
      ",name
CREATE #path,"company_name"
PUT #path,name
CLOSE #path
```



```

PROCEDURE input_items
PRINT CHR$(12)
PRINT TAB(20); "I N P U T \
S E E I T E M S "
PRINT \ PRINT
PRINT TAB(25); "Code"; "
"; "Name"
TYPE types_all=code:
STRING[4]; name:STRING[20]
DIM x: INTEGER
DIM choice:STRING[20]
x=0
DIM path:BYTE
DIM types:types_all
OPEN #path,"items_list"
WHILE NOT(EOF(#path)) DO
50 GET #path,types
x=x+1
PRINT TAB(25); types.code; "
";
PRINT types.name
IF types.code=" " THEN
PRINT TAB(20); "You have
space for "; 31-x; "
Items"
PRINT
60 INPUT "Input CODE <4
characters> <e> to END ",
choice$
IF choice$="e" OR choice$="
E" THEN 100
IF LEN(choice$)>4 THEN 60
types.code=choice$
INPUT "Input Name <20
Characters> ",types.name
SEEK #path,(x-1)*SIZE(types)
PUT #path,types
ENDIF
GOTO 50
ENDWHILE
100 CLOSE #path
PROCEDURE create_items
PRINT CHR$(12)
DIM x: INTEGER
TYPE types_all=code:
STRING[4]; name:STRING[20]
DIM a:STRING[1]
DIM choice: INTEGER
PRINT " The first thing we
have to do is create a
file that contains"
PRINT " a list of Catagories
and Codes for your
Expenses and Deposits"
PRINT
PRINT " You have space for
30 ITEMS "
PRINT
PRINT " The idea is to
allocate a code to a
particular Expense "
PRINT " or Deposit item that
is easy to remember when
you have to "
PRINT " enter in your cheque
butts and deposit slips. "
PRINT
PRINT " For example, instead
of having a code system

```

```

such as :-"
PRINT " Code
Item"
PRINT " 1000
Hire of Equipment"
PRINT " 1001
Salary J Tanner"
PRINT " 1002
Tax J Tanner"
PRINT " 1003
Car lease"
PRINT
PRINT " and so on, "
PRINT
PRINT " It is much easier to
have a system like :-"
PRINT
PRINT " Code
Item"
PRINT " hire
Hire of Equipment"
PRINT " sajt
salary J Tanner"
PRINT " tajt
Tax J Tanner"
PRINT " carl
Car Lease"
PRINT
PRINT " The only restriction
on entering expense codes
is that they"
PRINT " have to be 4
characters long. It is
also easier to enter them"
PRINT " later if they are
all lower case. Remember
the computer will"
PRINT " not recognise your
code unless it is EXACTLY
the same as you"
PRINT " have coded it. For
example Hire is different
to hire"
PRINT
PRINT " Deposit items are
different in that the
first three characters"
PRINT " of the deposit code
HAS to be dep. (in lower
case)"
PRINT
PRINT " The fourth character
can be a letter or number"
PRINT " For example:- "
PRINT " dep1
Sale of sheep"
PRINT " dep2
Sale of hay"
PRINT " dep2
Profits from Stock"
PRINT
PRINT " Remember you only
have a total of 30 Spaces
in the file for"
PRINT " both Deposits and
Expenses HOWEVER "
PRINT
PRINT " You do NOT have to
enter them all here, but

```

```

can enter them "
PRINT " later, even when you
are entering cheques and
deposits"
PRINT \ PRINT \ PRINT \
PRINT
INPUT " Press <enter> to
Begin to enter codes",a
DIM path:BYTE
DIM types:types_all
CREATE #path,"items_list"
choice=30
types.code=" "
types.name=" "
FOR x=1 TO choice
PUT #path,types
NEXT x
CLOSE #path

```

PLEASE NOTE

that lines indented by two (2) characters are in fact a continuation of the previous line.

```

PROCEDURE input_items
PRINT CHR$(12)
PRINT TAB(20); "I N P U T \
S E E I T E M S "
PRINT \ PRINT
PRINT TAB(25); "Code"; "
"; "Name"
TYPE types_all=code:STRING
[4]; name:STRING[20]
DIM x: INTEGER
DIM choice:STRING[20]
x=0
DIM path:BYTE
DIM types:types_all
OPEN #path,"items_list"
WHILE NOT(EOF(#path)) DO
50 GET #path,types
x=x+1
PRINT TAB(25); types.code; "
";
PRINT types.name
IF types.code=" " THEN
PRINT TAB(20); "You have
space for "; 31-x; "
Items"
PRINT
60 INPUT "Input CODE <4
characters> <e> to END
",choice$
IF choice$="e" OR
choice$="E" THEN 100
IF LEN(choice$)>4 THEN 60
types.code=choice$
INPUT "Input Name <20
Characters> ",types.name
SEEK #path,(x-1)*SIZE(types)
PUT #path,types
ENDIF
GOTO 50
ENDWHILE
100 CLOSE #path

```

****PLEASE NOTE****

that lines indented by two (2) characters are in fact a continuation of the previous line.#####

```
PROCEDURE create_start_bal
DIM path:BYTE
DIM begin_bal:REAL
begin_bal=0
CREATE #path,"start_bal"
PUT #path,begin_bal
CLOSE #path
```

PRINT_CASH****PLEASE NOTE****

that lines indented by (2) two characters are in fact a continuation of the previous line.

```
PROCEDURE search_month_p
PRINT CHR$(12)
PRINT "**** P R I N T I N G
      T O P R I N T E R ****"
TYPE types_all=code
      :STRING(4);name:STRING(20)
DIM types(30):types_all
TYPE record=date:STRING(8);
      number:INTEGER;payee:
      STRING(25); ammt:REAL;
      item:STRING(4)
DIM chq_rec:record
DIM type_total(30):REAL
DIM pathp:BYTE
FOR x=1 TO 30
type_total(x)=0
NEXT x
RUN items(types)
DIM path:BYTE
DIM choice:STRING(1)
DIM search:STRING(2)
DIM count:INTEGER
DIM total:REAL
DIM total2:REAL
DIM deposits:REAL
total=0
total2=0
count=0
deposits=0
chq_rec.date=""
chq_rec.number=0
chq_rec.payee=""
chq_rec.ammt=0
chq_rec.item=""
INPUT "What Month to PRINT
      <eg 06 for June>?",search
IF LEN(search)<2 THEN
search="0"+search
ENDIF
PRINT CHR$(12)
RUN macau_title
RUN title_p
```

```
OPEN #path,"chq_record"
WHILE NOT(EOF(#path)) DO
GET #path,chq_rec
IF MID$(chq_rec.date,4,2)
=search THEN
=search THEN
IF LEFT$(chq_rec.item,3)
="dep" THEN
deposits=deposits+chq_rec
.ammt.
RUN print_chq_p(chq_rec)
ELSE
total=total+chq_rec.ammt
FOR x=1 TO 30
IF chq_rec.item=types(x)
.code THEN
type_total(x)=type_total(x)
+chq_rec.ammt
ENDIF
NEXT x
count=count+1
RUN print_chq_p(chq_rec)
ENDIF
ENDIF
ENDWHILE
OPEN #pathp,"/p"
PRINT #pathp,
PRINT #pathp,TAB(18);
"Totals ";
PRINT #pathp,TAB(39);"";
PRINT #pathp USING "r12.2^";
deposits;
CLOSE #path
PRINT #pathp,TAB(53);"";
PRINT #pathp USING "r12.2^";
total
PRINT #pathp
FOR x=1 TO 30
IF type_total(x)<>0 THEN
total2=total2+type_total(x)
PRINT #pathp,TAB(18);"";
PRINT #pathp,types(x).name;
PRINT #pathp,TAB(53);"";
PRINT #pathp USING "r12.2^";
type_total(x)
ENDIF
NEXT x
PRINT #pathp,
PRINT #pathp,TAB(53);"";
PRINT #pathp USING "r12.2^";
total2
PRINT #pathp,
PRINT #pathp,CHR$(12)
CLOSE #pathp
PROCEDURE print_chq_p
TYPE record=date:STRING(8)
      ;number:INTEGER; payee
      :STRING(25); ammt:REAL; item
      :STRING(4)
PARAM chq_rec:record
DIM pathp:BYTE
OPEN #pathp,"/p"
PRINT #pathp,chq_rec.date;
PRINT #pathp,TAB(11);chq_rec
.number;
PRINT #pathp,TAB(18);chq_rec
.payee;
IF LEFT$(chq_rec.item,3)<>
"dep" THEN
```

```
PRINT #pathp,TAB(55);"";
ELSE PRINT#pathp,TAB(41);"";
ENDIF
PRINT #pathp USING "r10.2^";
chq_rec.ammt;
PRINT#pathp,TAB(68);chq_rec
.item
CLOSE #pathp
END
PROCEDURE title_p
DIM top_title:STRING(64)
DIM pathp:BYTE
OPEN #pathp,"/p"
PRINT #pathp,"Date";
PRINT #pathp,TAB(10);"Chq
No";
PRINT #pathp,TAB(18);"Payee/
Depositor";
PRINT #pathp,TAB(42);
"Deposits";
PRINT #pathp,TAB(56);
"Payments";
PRINT #pathp,TAB(68);"Item"
CLOSE #pathp
END
PROCEDURE macau_title
DIM co_name:STRING(20)
DIM pathp:BYTE
OPEN #pathp,"company_name"
GET #pathp,co_name
CLOSE #pathp
OPEN #pathp,"/p"
PRINT #pathp,TAB(15),co_name
PRINT #pathp,
CLOSE #pathp
PROCEDURE items
TYPE types_all=code
      :STRING(4);name:STRING(20)
DIM path:BYTE
PARAM types(30):types_all
DIM x:INTEGER
OPEN #path,"items_list"
FOR x=1 TO 30
GET #path,types(x)
NEXT x
CLOSE #path
```

****PLEASE NOTE****

that lines indented by two (2) characters are in fact a continuation of the previous line

```
PROCEDURE year_print
PRINT CHR$(12)
PRINT "**** Y E A R P R I N T
      O U T ****"
TYPE months_items=month
      :STRING(10); ammt(30):REAL;
      month_tot:REAL
TYPE record=date:STRING(8);
      number:INTEGER; payee
      :STRING(25); chq_ammt
      :REAL; item:STRING(4)
DIM chq_rec:record
DIM year_tot(12):months
```

```

_items
TYPE it=sys_code:STRING(4);
  sys_name:STRING(20)
DIM sys_item(30):it
DIM work_rec(12):STRING(10)
DIM items_tot(30):REAL
DIM y:INTEGER
DIM gr_tot:REAL
DIM gr_tot2:REAL
DIM gr_tot3:REAL
gr_tot:=0
gr_tot2:=0
gr_tot3:=0
FOR y=1 TO 30
  items_tot(y)=0
NEXT y
DIM path:BYTE
DIM x:INTEGER
DIM temp:INTEGER
FOR x=1 TO 12
  year_tot(x).month_tot=0
FOR y=1 TO 30
  year_tot(x).amnt(y)=0
NEXT y
NEXT x
FOR x=1 TO 12
  RUN months(work_rec)
  year_tot(x).month
    =work_rec(x)
NEXT x
x=0
OPEN #path,"items_list"
WHILE NOT(EOF(#path)) DO
  x=x+1
  GET #path,sys_item(x)
ENDWHILE
CLOSE #path
OPEN #path,"chq_record"
WHILE NOT(EOF(#path)) DO
  GET #path,chq_rec
  IF LEN(chq_rec.date)=8 THEN
    temp=VAL(MID$(chq_rec.date,
      4,2))
    FOR y=1 TO 30
      IF sys_item(y).sys_code
        =chq_rec.item THEN year
        _tot(temp).amnt(y)=year
        _tot(temp).amnt(y)+chq
        _rec.chq_amnt
    ENDIF
  NEXT y
ENDIF
ENDWHILE
CLOSE #path
OPEN #path,"/p"
PRINT #path,CHR$(27);
  CHR$(20)
FOR x=1 TO 12
  PRINT #path,TAB(6+x*10);
    LEFT$(year_tot(x).month
      ,3);
NEXT x
PRINT #path,
FOR y=1 TO 30
  IF sys_item(y).sys_code<>" "
    THEN
  IF LEFT$(sys_item(y).sys
    _code,3)<>"dep" THEN
  PRINT #path,LEFT$(sys

```

```

  _item(y).sys_name,10);
FOR x=1 TO 6
  IF year_tot(x+6).amnt(y)<>0
    THEN
  PRINT #path,TAB(2+x*10);
  PRINT #path USING "r9.2>";
    year_tot(x+6).amnt(y);
    year_tot(x+6).month_tot;
    =year_tot(x+6).month_tot
    +year_tot(x+6).amnt(y)
    items_tot(y):=items_tot
    (y)+year_tot(x+6).amnt(y)
  ENDIF
NEXT x
FOR x=1 TO 6
  IF year_tot(x).amnt(y)<>0
    THEN
  PRINT #path,TAB(62+x*10);
  PRINT #path USING "r9.2>";
    year_tot(x).amnt(y);
    year_tot(x).month_tot;
    =year_tot(x).month_tot+
    year_tot(x).amnt(y)items
    _tot(y):=items_tot(y)+
    year_tot(x).amnt(y)
  ENDIF
NEXT x
PRINT #path,

```

```

PRINT #path
PRINT #path,CHR$(27);
  CHR$(19)
FOR y=1 TO 30
  IF sys_item(y).sys_code<>" "
    THEN
  IF LEFT$(sys_item(y).sys
    _code,3)<>"dep" THEN
  PRINT #path,sys_item(y).sys
    _name;
  PRINT #path,TAB(35);
  PRINT #path USING "r10.2>";
    items_tot(y)
  gr_tot2=gr_tot2+items_tot(y)
  ENDIF
ENDIF
NEXT y
PRINT #path
PRINT #path,"*** T O T A L
  CHEQUES ***";
PRINT #path,TAB(35);
PRINT #path USING "r10.2>";
  gr_tot
PRINT #path
PRINT #path
FOR y=1 TO 30
  IF sys_item(y).sys_code<>" "
    THEN

```

You will note that included with this article, is a list of the procedures needed for this program to work. It is important that the procedures listed are 'SAVED' under their correct titles.

```

ENDIF
ENDIF
NEXT y
FOR y=1 TO 30
  IF LEFT$(sys_item(y).sys
    _code,3)="dep" THEN
  FOR x=1 TO 6
    items_tot(y)=items_tot(y)
    +year_tot(x+6).amnt(y)
  items_tot(y)=items_tot(y)
    +year_tot(x).amnt(y)
  NEXT x
ENDIF
NEXT y
PRINT #path
PRINT #path,"";
FOR x=1 TO 6
  PRINT #path,TAB(2+x*10);
  PRINT #path USING "r9.2>";
    year_tot(x+6).month_tot;
    gr_tot=gr_tot+year_tot
    (x+6).month_tot
  NEXT x
FOR x=1 TO 6
  PRINT #path,TAB(62+x*10);
  PRINT #path USING "r9.2>";
    year_tot(x).month_tot;
    gr_tot=gr_tot+year_tot(x)
    .month_tot
  NEXT x
PRINT #path
PRINT #path

```

```

  IF LEFT$(sys_item(y).sys
    _code,3)="dep" THEN
  PRINT #path,sys_item(y).sys
    _name;
  PRINT #path,TAB(35);
  PRINT #path USING "r10.2>";
    items_tot(y)
  gr_tot3=gr_tot3+items_tot(y)
  ENDIF
ENDIF
NEXT y
PRINT #path
PRINT #path
PRINT #path,"*** T O T A L
  DEPOSITS ***";
PRINT #path,TAB(35);
PRINT #path USING "r10.2>";
  gr_tot3
PRINT #path,CHR$(12)
CLOSE #path
PROCEDURE months
DATA "July","August",
  "September","October",
  "November","December",
  "January","February",
  "March","April","May",
  "June"
DIM x:INTEGER
PARAM months(12):STRING(10)
FOR x=1 TO 12
  READ months(x)
NEXT x continued on page 62

```

A*FORTH

- A Review

by Mike Turk

A*FORTH

When I was at CoCoConf 86 I saw John Redmond's A*FORTH (Version 3.69) being demonstrated by the author. It was an impressive demonstration of a most impressive Australian product.

A HOST OF MIND BLOWING FEATURES

This implementation of the FORTH language has a host of mind-blowing features! It loads from disk or tape and the disc version (3 are provided) comes complete with source code. The FORTH environment comes complete with its own editor, high resolution graphics language and virtual memory manager.

The FORTH screens are displayed using a high resolution text mode similar to that of Telewriter 64. However, you can customize your character set, screen width, left and right margins and scroll protect the top and bottom of your screen dynamically.

GRAPHICS PLUS

There are 11 graphic modes (including 8 colour modes and 3 semigraphic modes), and you can mix text and graphics. The graphics commands are simple and common sense; for example FINE COLOR GRAPHICS or COARSE SEMIGRAPHICS set the mode. You can use PAPER and INK to set the background and foreground colors and in addition to commands like CIRCLE and LINE there are turtle graphics commands as well!

A SPEED DEMON

Another feature of A*FORTH is SPEED. It is within 3 times the speed of assembler or machine code and 30 to 40 times the speed of interpreted BASIC. FORTH is a handy language to develop programs in, you can even define your own vocabulary in terms of the FORTH standard words and any other words that may have been previously been defined. The editor is easy to use and the virtual memory implementation straightforward as you write and modify programs.

USEFUL

FORTH can do almost anything. You can write an adventure game, a database manager, word processor, graphics utilities (Graphicom 1 was written using FORTH), or games using color which run fast. There are implementations of other languages written using FORTH such as BASIC, LISP, PASCAL and PROLOG. In the unlikely event that a program you developed using FORTH did not run quickly enough there is a built-in assembler. It is also a language for fast development. Why not develop your next program using FORTH?

COMPACT

FORTH code takes up almost no space - in fact it is more compact than assembler (this is because it uses a threaded list for its dictionary).

DOCUMENTATION

John's implementation of FORTH is based on the language described in Brodie's "Beginning Forth", which is (as far as A*FORTH is concerned the standard). The manual that comes with the package is primarily concerned with the implementation of FORTH provided - it is not a FORTH textbook.

THE LANGUAGE ITSELF

FORTH uses stacks to manipulate its data. If you have used a Hewlett-Packard Calculator you will have used a stack. The examples of FORTH given in the recent articles by Messrs Poxon and Redmond are fairly advanced, so do not be put off. The FORTH practitioners assure me that it is easy to learn, fast to write, fast to execute and easy to debug.

BUY AUSTRALIAN

John Redmond's A*FORTH (Aussie FORTH) is a well written, compact product, it produces fast efficient code. At \$60 it represents very good value for money. If our local authors are not supported then what incentive is there for them to put the hours and hours of their own time into developing their products? A*FORTH is obtainable from the suppliers advertised from time to time in Australian CoCo Magazine and Softgold Magazine.

FORTH

CoCo: Forth Language

by Peter Spowart

I HOPE THIS ARTICLE is of interest. It's a double-precision version of the quadratic solver A*FORTH source from October's Australian Rainbow. As you see, it's got quite long, but could easily be made to look easier with shorter names for the variables.

The limit of 32767 still exists for input. The first decimal place is accurate, but the second place is a bit iffy. Most of the time it is just truncated but now and again it rounds up, as there was no attempt made to direct the outcome.

As per John Poxon's routine, it accepts integers only, and will read negative signs. I have assumed that the solver equation holds good with '+' and or '-' signs on A, B and C. I confess that this program outputs negative real parts could be all wrong!

The SQRT routine was originally tried with three decimal places, but this worked out very slow because of the simple nature of UD/, which is successive subtractions....

There, I have done with the apologies for its shortcomings, it actually works. During the weeks it took me to get it together, I had some doubts. Thanks are due to John Poxon for suggesting an interesting exercise. I now know much more about double numbers than I did before!

On another topic, you will recall that John Redmond put out some graphics routines with A*FORTH before XGAL came out in Australian Rainbow. As I wanted to play around with Turtle routines at the time, I sort of pushed everything of John's together in high level Forth (which is all I know), and this works too. I could easily send details of what I did if Rainbow readers are interested. I wonder if the current disk version of A*FORTH has got all this wrapped up in a much neater way!

The Listing:

```
VARIABLE PRECISION
: 2CONSTANT ( D-- ) CREATE , ,
DOES> ( --D ) 2@ ;
: 2VARIABLE ( -- ) CREATE 0 ,
0 , DOES> ( --ADDR ) ;
: .PLACES PRECISION @ 0 DO #
LOOP ASCII . HOLD ;
: .REAL SWAP OVER DABS <#
.PLACES #S SIGN #> TYPE SPACE ;
2VARIABLE BSQD
2VARIABLE 4AC
2VARIABLE 2A
2VARIABLE NEGB
2VARIABLE SCALER
2VARIABLE DIVIDER
2VARIABLE COUNTER
2VARIABLE ORIGINAL
2VARIABLE TARGET
2VARIABLE GUESS
VARIABLE 'DIVIDE
VARIABLE ASIGN
VARIABLE BSIGN
VARIABLE CSIGN
VARIABLE ACSIGN
VARIABLE DENSIGN
: @EXECUTE ?DUP IF
EXECUTE THEN ;
```

```
: M*/' TARGET 2@ GUESS 2@
DROP 1 SWAP M*/ 2DUP
COUNTER 2! ;
: D>- D< NOT ;
: DO< DROP 0< ;
: UD/ 0. COUNTER 2! TARGET 2@
ORIGINAL 2! BEGIN TARGET 2@
GUESS 2@ D- 2DUP TARGET 2! 0<
NOT WHILE COUNTER 2@ SWAP DUP
32767 - IF SWAP 1. D+ ELSE 1+
SWAP THEN COUNTER 2! DROP
REPEAT DROP COUNTER 2@ ;
: 10/ 2@ 1 10 M*/ ;
: DOWN-SHIFT ORIGINAL
10/ TARGET 2! DIVIDER 2@
GUESS 2! ;
: SCALER-DOWN SCALER 2@ 999.
D>- IF SCALER 10/ SCALER 2!
DIVIDER 10/ DIVIDER 2!
DOWN-SHIFT THEN ;
```

```
: SELECT-DIVIDE GUESS 2@
32768. D>- IF ['] UD/
'DIVIDE ! ELSE ['] M*/'
'DIVIDE ! THEN 'DIVIDE @
@EXECUTE ;
: RE-CALC COUNTER 2@ GUESS 2@
D+ 1 2 M*/ GUESS 2!
ORIGINAL 2@ TARGET 2! ;
: TEST-UNEQUAL GUESS 2@
D- DABS 2. D>- ;
: 3S 3 SPACES ;
: A. 3S ." A = " ROT DUP . ;
: B. 3S ." B = " ROT DUP . ;
: C. 3S ." C = " ROT DUP . ;
: LISTVAR A. B. C. CR CR ;
: S>D DUP 0< NEGATE ;
: SQBDO SWAP DUP DUP * S>D
BSQD 2! SWAP ;
: 4ACDO ROT DUP ROT * 4 * S>D
```

```
SWAP OVER DABS 4AC 2! ACSIGN
! ;
: 2ADO 2* S>D SWAP OVER DABS
2A 2! ASIGN ! ;
: BDO S>D SWAP OVER DABS NEGB
2! BSIGN ! ;

: 1CALCS SQBDO 4ACDO 2ADO BDO ;
: COMPLEX? ACSIGN @ -1 - IF
1. ELSE BSQD 2@ 4AC 2@ D-
THEN ;
: B@100* NEGB 2@ 100 1 M*/ ;
: SQRT 10.000 SCALER 2! 1.000
DIVIDER 2! 2 PRECISION !
SCALER 2@ ROT 1 M*/ 2DUP
ORIGINAL 2! TARGET 2! DIVIDER
2@ GUESS 2! BEGIN
SELECT-DIVIDE TEST-UNEQUAL
WHILE RE-CALC GUESS 2@ 32768.
D>- IF SCALER-DOWN THEN REPEAT
COUNTER 2@ ;
: TITLE PAGE CR
ROOTS OF QUADRATIC EQUATION"
CR CR ;
: NGETROOT BSQD 2@ 4AC 2@
ACSIGN @ -1 - IF ELSE DNEGATE
THEN D+ DROP ABS SQRT ;
: CGETROOT 4AC 2@ BSQD 2@
DNEGATE D+ DROP ABS SQRT ;

: NIROOT BSIGN @ -1 - IF
B@100* DABS ELSE B@100*
DNEGATE THEN NGETROOT D+ SWAP
OVER DABS TARGET 2! DENDSIGN !
2A 2@ DABS GUESS 2! UD/
DENDSIGN @ -1 - IF DNEGATE
THEN 0 DENDSIGN ! ASIGN @ -1
- IF DNEGATE THEN 3S
." 1ST REAL - " .REAL ;
: N2ROOT BSIGN @ -1 - IF
B@100* DABS ELSE B@100*
DNEGATE THEN NGETROOT D- SWAP
OVER DABS TARGET 2! DENDSIGN !
2A 2@ DABS GUESS 2! UD/
DENDSIGN @ -1 - IF DNEGATE
THEN 0 DENDSIGN ! ASIGN @ -1
- IF DNEGATE THEN 3S
." 2ND REAL - " .REAL ;
: LEFT-SIDE B@100* DABS TARGET
2! 2A 2@ DABS GUESS 2! UD/
ASIGN @ -1 - IF DROP ELSE
DNEGATE THEN ;

: RIGHT-SIDE CGETROOT TARGET
2! 2A 2@ DABS GUESS 2! UD/ ;
: C1ROOT LEFT-SIDE 3S
." 1ST COMPLEX - " .REAL
RIGHT-SIDE ." + " .REAL
." I " ;
: C2ROOT LEFT-SIDE 3S
." 2ND COMPLEX - " .REAL
RIGHT-SIDE ." - " .REAL
." I " ;
: NONCOMPLEX NIROOT CR
N2ROOT ;
: ISCOMPLEX C1ROOT CR C2ROOT ;
: Q TITLE LISTVAR 1CALCS
COMPLEX? DO< IF ISCOMPLEX ELSE
NONCOMPLEX THEN ;
```

□

COLOR and PMODE 3/4 CHANGER

Hardware Modification

by Paul Kuhn

THE PMODE 3/4 SWITCH lets you switch between, (as it says) PMODE3 and PMODE4. A word of warning though - ONLY when you are already in one of those modes. Changing the switch in other modes will make strange things occur on your screen, but rest assured you will not damage anything by doing so; so have a look anyway.

This project uses another gate of the exclusive OR (XOR) to do its task in a similar fashion to that of the alternative colour set switch. The advantage of these two modifications is that you can see all four variations possibly available to you for any particular program without having to delve into the software itself. (If it's a 32K program it could take a while in software to find every &HFF22 and to make sure it's the relevant one at that!)

The colour changer switch does just that - changes the screen to the only alternative colour set that the CoCo allows (the video generator actually). THIS one will operate in ANY mode, graphics, semi-graphics and alpha-numeric normal or inverse. Each mode (as above) has two colour sets available via the CSS pin on the VDG and this is what is used here.

First, some explanation of what we are doing and some basics in the CoCo.

In the supplied diagram is a copy of the Truth Table for this chip we are using, the SN74LS86, a QUAD 2-input EXCLUSIVE OR GATE. A zero (0) in the table represents a logic 0 or in real terms that means ground, earth or voltage reference point. In the CoCo there are plenty of places to pick up a ground. A one represents a logic 1 which in real terms is actually a positive 5 volts (+5V) with respect to the voltage reference point, in this case, ground.

I have drawn the circuit so that the switches switch between ground and +5V, labelled NORMAL and ALTERNATIVE. From the Truth Table you can see that when IN/A

is 0, the OUT remains the same as the IN/B which is connected to the original point in the CoCo in both cases. Therefore, from the Truth Table, you can also see that when the switch is placed to the +5V, or LOGIC 1 state, then the OUT of the Truth Table is the INVERSE of the IN/B, (connected to the original point in the CoCo).

The Truth Table merely tells you what to expect from a device for a given output with a given input on the input leads. This is called Transistor Transistor Logic (TTL).

Looking at the table and the LOGIC diagram of the chip itself the IN/A and IN/B referred to in the Truth Table is actually pins 1 and 2 of the first separate device in the chip. Pin 3 is the OUTPUT, labelled OUT in the Truth Table. Therefore, pins 1, 2 and 3 are ONE complete TTL exclusive OR gate. Power to these four individual gates is supplied internally in the chip from the VCC pin 14 (standard pin configuration) and GND on pin 7.

Getting back to our device, this now becomes the basic element, along with switches, for our two changers.

I shall now refer to pins and chips directly from the drawing.

Label:	Description:	Chip(I/C)	Pin #
*	Denotes ACTIVE LOW(logic 0 to work)		
*INT/EXT	Internal/External Alpha set.	VDG	31
GMO	Graphics Mode 0	VDG	30
CSS	Colour Set SELECT	VDG	39
PB4	Data line 5	PIA	14
PB3	Data line 4	PIA	13

Let's say we are in PMODE 4, color set Black and White, with a white border. When both switches are in the normal position, we will get white border, black and white picture. XOR2-(logic)0, XOR1-1 therefore XOR3-1 (refer to the Truth Table), applies a logic 1 to the CSS-39. To change the screen picture to the alternative colour set available in that mode, green everything (yech), operate the colour set switch to

+5V. Now XOR1-1 (still) and XOR2-1 giving a REVERSED output at XOR3, in fact a logic 0 that is sent to the VDG which sends out the alternative colour set!

Very Simple! The PMODE 3/4 switch operates in the same manner. This is the best way to do all of this so that when the switches are NORMAL, all the screen displays are as the software dictates, as normal. Easy.

Doing it

I would suggest that you should be relatively conversant with your soldering iron and tools. Do NOT use a lead burning soldering iron as the tracks to cut and solder are very narrow. I used two switches that are SPST (Single Pole Single Throw) as shown in the diagram. Some people refer to them as change over switches (CO switch) like me.

Parts list and requirements

- o 74LS86 - 2 input quad exclusive OR gate. About \$1.40.
- o CO switches - 2 of, I used sub-miniature toggles.
- o Small piece of vero board (experimenters board).
- o Small diameter insulated

socket. (Use the best sockets - it pays).

- o 2 inches of electrical tape.
- o Solder.

Requirements.

- o Delicate soldering Iron, weller stations are good.
- o Diags. (small ones of course).
- o Small sharp knife to cut tracks
- o Multimeter with X1 scale (at

1.5V) in OHMS range.
o Care and attention.

Now onto the job!

1. Mount 14 pin socket onto vero board with a good foot of wire to pin numbers 1, 2, 3, 4, 5, 6, 7 & 14.

2. Insert 74LS86 the right way round.

3. Mount switches in CoCo in easy to get at place.

4. Take wire from pin 2 of 74LS86 to centre of one switch.

4a. Take wire from pin 5 of 75LS86 to center of other switch.

5. Run another wire from one side of switch to CoCo ground.

5a. Run another wire from other side of switch to CoCo +5 volts.

6. Run ground and +5V to other switch from first switch the same.

7. Locate the VDG (6874), and the other two PIA's (6821 or 6822).

8. Put meter on OHMS X1 scale and put one lead on VDG pin 39.

9. Check which PIA pin 13 is dead short to it.

10. Locate the track between the two pins by pushing one probe into the most likely track and continue until you find it.

11. Cut that track at a point where you can get your soldering iron in comfortably to both sides of the cut track.

12. Check again from VDG 30 or 31 to PIA pin 14 and cut that track likewise.

13. Solder the wire from pin (XOR)1 to the track that goes to PIA pin 13.

14. Solder the wire from pin 3 to the other side of that cut track (that goes to VDG pin 39).

15. Solder the wire from pin 4 to the track that goes to PIA pin 14.

16. Solder the wire from pin 6 to the other side of that cut track (that goes to VDG pin 30 & 31).

17. Connect the ground wire

(pin 7) to ground.

17a. Connect the VCC wire (pin 14) to +5 volts.

18. Test for shorts where you have been working with meter.

19. That's it.

A word on the book

The green book by Tandy called a technical reference manual.

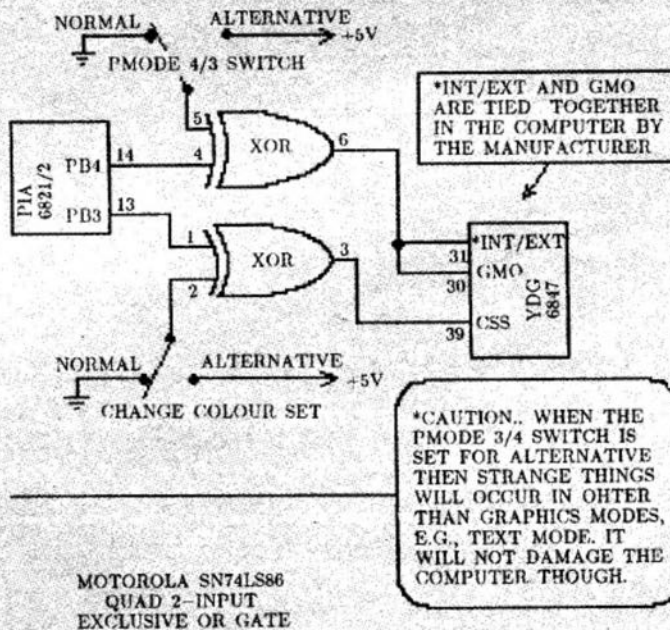
On page 24 there's a table, #1, called the MODE SELECTION. It doesn't give the whole details. Well, almost. Anyway, pencil this in somewhere on the page.

PIA Register Bits (at address \$FF22 - decimal 65314)
 Bit 3 - is the colour set select line, 0 or 1, to the VDG
 Bit 4 - is GMD
 Bit 5 - GM1
 Bit 6 - GM2
 Bit 7 - *A/G

Data bits Coloumn, bit 6 - INV (inv. select in alphanumeric)
 Data bits coloumn, bit 7 - *INT/EXT

The last row, 256 x 192 Graphics is PMODE 4
 Next row up, 128 x 192 Colour Graphics is PMODE 3.
 Next row up, 128 x 192 Colour Graphics is PMODE 2.
 Next row up, 128 x 96 Colour Graphics is PMODE 1.
 Next row up, 128 x 96 Colour Graphics is PMODE 0.
 Next row up is not used by Tandy as such.

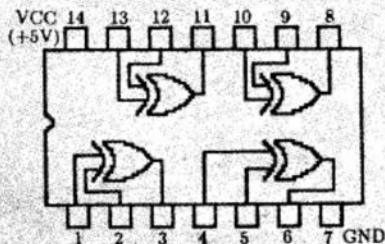
COLOUR 7 PMODE 3/4 CHANGER
 PAUL KUHN, 3 BAY ST., SEACOMBE GARDENS, 5047
 THESE MODIFICATIONS ALLOW YOU TO HARDWARE
 'FLIP' BETWEEN PMODE 4 AND PMODE 3 AT WILL
 AND TO ALSO SEE THE DIFFERENT COLOUR SETS
 AVAILABLE IN ANY MODE.



MOTOROLA SN74LS86
 QUAD 2-INPUT
 EXCLUSIVE OR GATE

TRUTH TABLE

IN		OUT
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0



CASHBOOK

continued from page 57

```
PROCEDURE correct_items
PRINT CHR$(12)
PRINT "***** CORRECT ITEMS
LIST *****"
DIM x: INTEGER
TYPE types_all=code:STRING
[4]; name:STRING[20]
DIM choice:STRING[4]
DIM path:BYTE
DIM types:types_all
x=0
OPEN #path,"items_list"
WHILE NOT(EOF(#path)) DO
GET #path,types
IF types.code(">") THEN
x=x+1
PRINT types.code;
PRINT " ";
PRINT types.name
ENDIF
ENDWHILE
PRINT "Number of Items "; x
INPUT "Input code of Item to
change ",choice
x=0
SEEK #path,0
WHILE NOT(EOF(#path)) DO
GET #path,types
x=x+1
IF types.code=choice THEN
PRINT types.code; " ";
types.name
PRINT
INPUT "Input New Code
",types.code
PRINT
INPUT "Input New Catagorie
",types.name
SEEK #path,(x-1)*SIZE(types)
PUT #path,types
ENDIF
ENDWHILE
CLOSE #path
```

HINT.....

POKEs

New POKES for 6ms second step rate and double sided drive access for the CoCo 3.

```
POKE &HD7C0,0 6ms
POKE &HD016,&H14 6ms
POKE &HD09F,&H41 Double Sided
POKE &HD0A0,&H42 Double Sided
POKE &HD7C0,&H2 Tandy drive
20ms Step rate
```

That's.....

LIFE

16K ECB APPLICATION

by Craig Stewart

LIFE - YET ANOTHER well worn version of "The Game of Life" based on rules by Conway. Being in ML and in 10-res it is easy and fast, giving the appearance of animation when run continuously. After a long (or short) while, the life will reach a stable pattern or die out all together.

It is a slightly messy and very simple version, in that it is 'display' only, giving no indications of how many generations have passed and how much life is left.

Also, it is not entirely accurate as the life is allowed to run off the screen and join the other side. Nonetheless it comes up with some interesting patterns and is a bit of fun to play around with.

It also includes a full screen editor for the life.

CONTROLS

<BREAK> - alternated between edit and run modes.

EDIT MODE

Cursor keys - move cursor around
<Space-Bar> - place life unit
C - Cancels life unit
<Clear> - clears all life units
S - Save screen in memory, if you want to keep pattern.

RUN MODE

<Space-Bar> - Continuous life form.
<Enter> - step by step life run.

The Listing:

```
1 GOTO5
2 '*LIFE-PROGRAM BY C. STEWART*
3 SAVE"146D:3":END
4 END
5 CLEAR 200, 15871
10 CLS:PRINT "loading data .. pl
ease wait..
20 LI=50:FOR X= 15872 TO 16252
STEP 70:LI=LI+10:T=0
30 READ T$:FOR Y=X TO X+69:READ
A$:A=VAL("&H"+A$):T=T+A:POKE Y,A
```

```
:NEXT Y:IF HEX$(T)=T$ THEN 40
35 PRINT "error in line ";LI:END
40 NEXT X:READ T$:T=0:LI=LI+10:F
OR Z=X-70 TO 16252:READ A$:A=VA
L("&H"+A$):T=T+A:POKE Z,A:NEXT Z
:IF HEX$(T)<>T$ THEN 35 ELSE PRI
NT:PRINT"data loading complete"
45 PRINT "press enter to save":L
INEINPUTZ$
50 SAVEM "LIFE", 15872, 16252, 1
5872 :END
60 DATA 1824,17,0,BC,16,0,E0,10,
8E,4,20,31,21,10,8C,5,DC,10,22,0
,60,34,20,A6,A4,81,AF,27,EE,81,4
F,10,27,0,22,81,60,27,2,20,E2,17
,0,73,35,20,8C,2,CD,26,9,86,4F,A
7,A9,B,B8,16,FF,CF,86,60,A7,A9,B
,B8,16,FF,C6,17,0
70 DATA 1AE2,57,35,20,8C,2,CD,27
,14,8C,2,DE,27,F,8C,2,EF,27,13,8
C,3,0,27,E,8C,2,BC,2F,9,86,4F,A7
,A9,B,B8,16,FF,9F,86,60,A7,A9,B,
B8,16,FF,96,8E,4,0,8C,5,FF,22,8,
A6,89,B,E8,A7,80,20,F3,8D,4C,AD,
9F,A0,0,81,3
80 DATA 1F10,10,27,0,56,F6,1,59,
C1,FF,26,4,81,0,27,E9,16,FF,68,8
E,0,0,31,A8,DF,E6,A0,EB,A0,3A,E6
,A4,31,A8,1E,EB,A1,3A,E6,A4,31,A
8,1E,EB,A0,3A,E6,A0,EB,A4,3A,39,
8E,4,0,86,60,A7,89,B,B8,A7,80,8C
,6,0,22,3,20,F3,39
90 DATA 188F,8E,4,C,86,4C,A7,81,
86,49,A7,81,86,46,A7,81,86,45,A7
,81,39,8E,5,10,E6,84,86,80,A7,84
,1F,12,34,20,AD,9F,A0,0,E7,84,35
,20,81,8,26,2,30,1F,81,C,26,5,17
,FF,B7,20,DC,81,9,26,2,30,1,81,A
,26,3,30,88,20,81
100 DATA 170B,5E,26,3,30,88,E0,8
C,4,20,22,3,8E,5,10,8C,5,E0,25,3
,8E,5,10,81,53,26,12,8E,4,20,A6,
84,A7,89,F,A0,30,1,8C,5,DC,27,A4
,20,F1,81,52,26,12,8E,4,20,A6,89
,F,A0,A7,84,30,1,8C,5,DC,27,8E,2
0,F1,81,20,26,6
110 DATA 19E5,86,4F,A7,84,20,15,
81,43,26,6,86,60,A7,84,20,B,81,3
,26,5,E7,A4,16,FE,8F,E7,A4,16,FF
,6D,3F,1A,6D,5,10,93,D5,25,2,DC,
D5,DD,C5,D,D8,26,2,8D,50,35,6,4,
D8,25,5,10,93,D9,27,C,5C,C1,8,26
,4,4C,5F,84,7,7E
120 DATA C8A,86,4F,A7,84,20,15,8
1,43,26,6,86,60,A7,84,20,B,81,3,
26,5,E7,A4,16,FE,8F,E7,A4,16,FF,
6D,3F
```


WHAT'S ON THE BEST OF CoCoOz

Best of CoCoOz #1. EDUCATION

ROADQUIZ ROB WEBB
 HANGMAN ALEPH DELTA
 AUSTGEOG P. THOMAS
 SPELL IAN LOBLEY
 FRACTUT ROBBIE DALZELL
 ICOSA BOB WALTERS
 TAXMAN TONY PARFITT
 MARKET ALEPH DELTA
 TOWNQUIZ ROB WEBB
 ALFABETA RON WEBB
 TANK ADDITION DEAN HODGSON
 TABLES BARRIE GERRAND
 KIDSTUFF JOHANNA VAGG
 FLAGQUIZ ROB WEBB

Best of CoCoOz #2 part 1. 16K GAMES.

LE-PAS Wrongsoft
 COCOMIND STEVE COLEMAN
 OILSLICK JEREMY GANS
 CCMETEOR BOB THOMSON
 BATTACK JEREMY GANS
 PROBDICE BOB DELBOURGO
 CHECKERS J & J GANS
 PYTHON ?
 POKERMCH GRAHAM & MATTHEWS
 SPEEDMATH DEAN HODGSON
 LNDATTCK ALDO DEBERNARDIS
 INVADERS DEAN HODGSON
 RALLY TONY PARFITT
 FOURDRAW JOHANNA VAGG

Best of CoCoOz #2 part2. 32K GAMES.

TREASURE DAVISON & GANS
 MASTERMIND GRAHAM JORDAN
 ANESTHESIA MIKE MARTYN
 OREGON TRAIL DEAN HODGSON
 ADVENTURE STUART RAYNER
 SHOOTING GALLERY TOM DYKEMA
 GARDEN DAVE BLUHDORN
 YAHTZEE KEVIN GOWAN
 BATTLESHIP CHRIS SIMPSON
 ANDROMIDA MAX BETTRIDGE

Best of CoCoOz #3. UTILITIES

PAGER ?
 HI ALEX. HARTMANN
 SPOOL64K WARREN WARNE
 CREATITL BRIAN FERGUSON
 FASTEXT OZ-WIZ
 DATAGEN ROBIN BROWN
 SPEEDCTR PAUL HUMPHREYS
 PRNTSORT PAUL HUMPHREYS
 BIGREMS BOB T
 DIR PAUL HUMPHREYS
 COPYDIR THOMAS SZULCHA
 LABELLER J.D.RAY
 SCRPRIT TOM DYKEMA
 MONITOR BRIAN FERGUSON
 BEAUTY BOB T
 PCOPY B. DOUGAN
 RAMTEST TOM DYKLEMA
 DISKFILE B. DOUGAN
 LABEL F. BISSELING

Best of CoCoOz #4. BUSINESS.

HI ALEX. HARTMANN
 (Disk Directory manager)
 BANKSTAT BARRY HATTAM
 (Statement annal & store)
 INSURE ROY VANDERSTEEN
 (Analyse home contents)
 SPOOL64K WARREN WARNE
 (Printer spooler req 64K)
 2BC WARREN WARNE
 (Hold 2 sep progs in mem)
 DATABASE PAUL HUMPHREYS
 (THE tape database)
 RESTACC DUNG LY
 (Tape restaurant accounts)
 PRSPDSHT GRAHAM MORPHETT
 (Disk print out SPDSHEET)
 PERSMAN PAUL HUMPHREYS
 (Personal finance management)
 CC5 GRAHAM MORPHETT
 (Sales Invoicing-tape sys)
 COCOFILE BRIAN DOUGAN
 (Tape data base)
 DPMS PAUL HUMPHREYS
 (Disk Program Management Sys)
 40KGREY RAY GAUVREAU
 (40K Basic for grey 64K CoCo)
 TAXATION ?
 (Calc tax payable)
 SPDSHEET GRAHAM MORPHETT
 (Disk 22 column spreadsheet)
 ACS3 GREG WILSON
 (Multi disk data base)

Best of CoCoOz #5. ADVENTURES.

ADV 32K S. RAYNER
 QUEST TONY PARFITT
 LABYRINT JAMES REDMOND
 ADV SEAN LOWE
 CRYSTAL C & K SPRINGETT
 PRISON TIM ALTON
 OPALTON IAN CLARKE
 WIZARD DARRELL BERRY
 TREASURE C. DAVISON
 LOST ALEX. HARTMANN

Best of CoCoOz #6. PRESCHOOL.

ALPHABET STUART DAWSON
 HATDANCE JOHANNA VAGG
 AUSTSONG McDERMOTT FAMILY
 ADVANCE McDERMOTT FAMILY
 WALTZING McDERMOTT FAMILY
 TIMEKANG McDERMOTT FAMILY
 BAND McDERMOTT FAMILY
 KIDSTUFF JOHANNA VAGG
 MATCHER ?
 LETTERS JACK FINNEN
 BABYSIT JOHANNA VAGG
 SPELLING JOHANNA VAGG
 SPEEDTAB DEAN HODGSON
 10 FACES JOHANNA VAGG

Best of CoCoOz #7. GRAFIX.

LIL'COCO ANDREW WHITE
 THE ROOM H. FREDRIKSON
 BACK ST JOY WALLACE
 LOCO MIKE D'ESTERRE
 COCO ART SANDY McGREGOR
 KANGA JOHANNA VAGG
 THE BOAT SANDY McGREGOR
 SAD COCO F. BOLLE
 TOWER C.A. SYMS
 WINDYDAY SARAH LAW
 SAILING STEVE YOUNGBERRY
 OUTHOUSE STEVE YOUNGBERRY
 SMURF JOHANNA VAGG
 SUNSTATE STEVE YOUNGBERRY
 HELICOPT ANDREW WHITE
 MARTHA ANDREW WHITE
 BAD MOON STEVE YOUNGBERRY
 MCC J. WALLACE
 EAGLE ?
 BLASTER PAUL YOULD
 FOGHORN PAUL STEVENSON

Best of CoCoOz #8 GAMES — 16K

ALIEN STUART SANDERS
 QWERL DARRELL BERRY
 SHOOTOUT CRAIG STEWART
 SHUTTLE CRAIG STEWART
 FROG DARREN OTTERY
 FROGRACE TOM LEHANE
 KIMMAT TOM LEHANE
 GRANDPRI DOUG GREY
 WATERWAR JUSTIN LIPTON
 CATERPIL JUSTIN LIPTON
 DETECT WAL STEPHENSON
 BREAKOUT WHY/BILT

Best of CoCoOz #9 GAMES — 32K

TRIOMINO BOB DELBOURGO
 MATCHEM C. BARTLETT
 GO BOB DELBOURGO
 NARZOD MAX BETTRIDGE
 CHOMPER MAX BETTRIDGE
 POPBALL MAX BETTRIDGE
 LUDO WHY/BILT
 SABRE ANDREW SIMPSON
 MOVEABOUT KEVIN GOWAN
 JIGSAW C. BARTLETT
 LABYRINT JAMES REDMOND
 TANK C. STEWART

Best of CoCoOz #10. EDUCATION2.

METEOR DEAN HODGSON
 DRIVTEST ANDREW SIMPSON
 SALE JUSTIN LIPTON
 TABLES PAT KERMODE
 OPALTON IAN G. CLARKE
 CAPITAL LETTERS BOB HORNE
 TEST MATCH JEFF SHEEN
 SENT END BOB HORNE
 ESCAPE DEAN HODGSON
 RAILMATH BOB HORNE
 COUNTDOWN DEAN HODGSON
 WHATZIT BOB HORNE
 HOMOPHONE BOB HORNE
 COMPWORDS BOB HORNE

Best of #11 EDUCATION 3 — Disk only CHATWIN MANOR BOB HORNE

TAPE \$16 each

DISK \$16 each

GOLDSOFT

P.O. BOX 1742, SOUTHPORT. QLD. 4215 Phone (075) 510 015

Goldsoft Price list as at March, 1987

Please tick your requirements.

HARDWARE

CoCoConnection: \$206.00 ()
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: \$39.75 ()

SOFTWARE

Magazines, Tapes & Disks

Australian CoCo (Advanced Programs for your CoCo):

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

Softgold (Programs for your CoCo):

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

Gold Disk - Available Quarterly:

1 Month: \$16.00 ()

MicoOz Tape (MC-10 Programs for your Mico):

12 Months \$75.00 ()
6 Months \$42.00 ()
1 Month \$10.00 ()

The CoCo3 Tape/Disk:

1 - Tape: \$10.00 Disk: \$16.00 ()
2 - 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:	()	()
# 2 - Games 32K:	()	()
# 3 - Utilities:	()	()
# 4 - Business :	()	()
# 5 - Adventure:	()	()
# 6 - Preschool:	()	()
# 7 - Graphics :	()	()
# 8 - Games 16K:	()	()
# 9 - Games 32K:	()	()
#10 - Education:	()	()
#11 - Education:	-	()

Special offer: Buy any 2 "Best of" and
pay only \$28.00!

BRIC-A-BRAC

Blank Tapes: 12 @ \$18.00 ()
 (C-30) 1 @ \$ 2.00 ()
Tape Cases: 12 @ \$ 5.00 ()
Disks 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 (): - - -

Visa (): - - -

Mastercard (): - - -

Expiry Date:/.....

Authorised Amount: \$.....

Signed:

ACT:
 CARRERA NTH JOHN BURGER 062 58 3924
 CARRERA STH LES THURBON 062 88 9226

NSV:
 SYDNEY:
 BANKSTOWN CARL STERN 02 649 3793
 BLACKTOWN KEITH GALLAGHER 02-627-4627
 CARLINGFORD ROSKO MCKAY 02 624 3353
 CHATSWOOD BILL O'DONNELL 02 419 6081
 COLYTON HERMAN FREDRIKSSON 02 623 6379
 FAIRFIELD ARTH PITTARD 02 72 2881
 GLADESVILLE MARK ROTHWELL 02 817 4627
 HILLS DIST ARTHUR SLADE 02 622 8940
 HORNSBY ATHALIE SMART 02 845 8630
 INGLEBURN STEPHEN RIDGEWAY 02 605 7382
 KENTHURST TOM STUART 02 654 2178
 LEICHHARDI STEVEN CHICOS 02 560 6207
 or GORGE ECHEGARAY 02 560 9664
 LIVERPOOL LEONIE DUGGAN 02-607-3791
 MACQUARIE FIELDS
 BARRY DARTON 02 618 1909
 NINTO GRAHAM POLLOCK 02 603 5028
 SUTHERLAND IAN ANNABEL 02 528 3391
 SYDNEY EAST JACKY COCKINOS 02 344 9111
 ALBURY RON DUNGAN 060 43 1031
 AFMIDALE DOUG BARBER 067 72 7647
 BLAXLAND BRUCE SULLIVAN 047 39 3903
 BROKEN HILL TERRY MOONAN 080 88 2382
 CAMDEN KEVIN WINTERS 046.66.8068
 COFFS HARBOUR BOB KENNY 066 51 2205
 COOMA ROSS PRATT 0648 23 055
 COOPANONG GEORGE SAVAGE 049 77 1054
 COOTAMUNDRA CHERYL WILLIS 069 42 2264
 DENTILQUIN WAYNE PATTERSON 058 81 3014
 DUEBO GRAEME CLARKE 068 89 6549
 FORBES JHANNA 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 ROB HILLARD 066 24 3089
 LITHGOW DAVID BEPGER 063 52 2282
 MAITLAND BILL SNOW 049 66 2557
 MOREE ALF BATE 067 52 2465
 MUDGEE BRIAN STONE 063-72-1958
 NARROMINE GRAEME CLARKE 068 89 6549
 NEWCASTLE LYN DAWSON 049 49 8144
 NOWRA ROY LOPEZ 044 48 7031
 ORANGE JIM JAMES 063 62 8625
 PARKES DAVID SMALL 068 62 2682
 PORT MACQUARIE RON LALOR 065 83 8223
 SPRINGWOOD DAVID SEAMONS 047 51 2107

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

USER CONTACTS

TAHMOOR GARY SYLVESTER 046 81 9318
 UFFER HUNTER TERRY GRAVOLIN 065 45 1698
 UFALLA 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 2128
 CLAYFIELD JACK FRICKER 07 262 8869
 COLL'WOOD PK AND'W SIMPSON 07 288 5206
 IPSWICH MICK MURPHY 07 271 1777
 PINE RIVERS BARRY CLARKE 07 204 2806
 SOUTH WEST BOB DEVRIES 07 372 7816
 SANDGATE MARK MICHELL 07 269 3846
 SCARBOROUGH PETER MAY 07 203 6723
 WOODRIDGE BOB DEVRIES 07 375 3161
 AIRLIE BEACH GLEN EVANS 079 46 1264
 BIGGENDEN ALAN MENHAM 071 27 1272
 BOWEN TERRY COITON 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 MORPHEIT 075 51 0577
 GYMPIE BERT LLOYD 071 8219100
 HEFVY BAY LESLEY HORWOOD 071 22 4989
 MACKAY LEN MALONEY 079511333x782
 MARYBOROUGH JOHN EFFER 071 23 1369
 MT ISA JACK RAE 077 43 3486
 MURGOON PETER ANGEL 071 68 1628
 ROCKHAMPTON KEIRAN SIMPSON 079 28 6162
 TARA DEBBIE DORFIELD 074 65 3177
 TOOWOOMBA LEN GEBSEKOWSKI 076 35 8264
 TOWNSVILLE JOHN O'CALLAGHAN 077 73 2064

SA:
 ADELAIDE JOHN HAINES 08 278 3560
 NORTH STEVEN EISENBERG 08 250 6214
 MORPHEITVALE KEN RICHARDS 08 384 4503
 PORT NOARLUNGA ROB DALZELL 08 386 1647
 SEACOMBE HTS GLENN DAVIS 08 296 7477
 PORT LINCOLN BILL BOARDMAN 086 82 2385
 PORT PIRIE VIC KNAUVERHASE 086 32 1230
 WHYALLA MALCOLM PATRICK 086 45 7637

TAS:
 DEVONPORT JEFF BEST 004 24 6759
 HOPART BOB DELBOURGD 002 25 3896
 KINGSTON WIM DE PUIT 002 29 4950
 LAUNCESTON BILL BOWER 003 44 1584
 WYNARD ANDREW WYLLIE 004 35 1839

VIC:
 MELBOURNE:
 MELBOURNE CCC JOY WALLACE 03 277 5182
 DANDENONG DAVID HORRIGGS 03 793 5157
 DONCASTER JUSTIN LIPTON 03 857 5149
 FRANKSTON BOB HAYTER 03.783.9748
 HARPE WARREN LEIGH HAMES 03 704 6680
 NTH EASTERN PETER WOOD 03 435 2018
 MELTON MARIO GERADA 03 743 1323
 RINGWOOD IVOR DAVIES 03 758 4496
 SUNBURY JACK SMIT 03.744.1355
 SUNSHINE IAN BUTTRIS 03 314 3240
 UPR F'TREE GLY RORY DOYLE 03 758 2671
 BAINSDALE COLIN LEHMANN 051 57 1545
 BALLARAT MARK BEVELANDER 053 32 6733
 CHURCHILL GEOFF SPOWART 051 22 1389
 DAYLESFORD DANNY HEDJI 054 24 8329
 GEELONG DAVID COLLEN 052 43 2128
 MAFFRA MAX HUCKERBY 051 45 4315
 MOE JOSEPH HESTER 051 27 7817
 MUE JIMMY WELSH 051 27 6984
 MORNINGTON MICHAEL MONCK 03 789 7997
 MORWELL GEORGE FRANCIS 051 34 5175
 SALE BRYAN MCHUGH 051 44 4792
 SHEPPARTON ROSS FARRAR 058 25 1007
 SMYTHSDALE TONY PATTERSON 053 42 6815
 SWAN HILL BARRIE GERRAND 050.32.2838
 TONGALA TONY HILLIS 058 59 2251
 TRARALGON LEIGH DAVIES 051 74 5552
 WORTHAGGI LOIS O'NEARA 056 72 1593
 YARRAVONGA KEN SPONG 057 44 1488

WA:
 PERTH IAIN MACLEOD 09 448 2136
 GIRRAWHEEN HANK WILLEMSSEN 09 342 7639
 KALGOORLIE TERRY BURNETT 090.21.5212

CANADA - CoCo:
 Ontario Richard Hobson 416 293 2346

SPECIAL INTEREST GROUPS

TEACHERS' INTEREST GROUP
 BRISBANE BOB HORNE 07 281 8151

BUSINESS:
 BRIZBIZ BRIAN BEPE-STREETER 07 349 4696

899 GROUPS:
 NATIONAL OS9 USERS' GROUP
 GRAEME NICHOLS 02 451 2954

NSW
 SYDNEY
 BANKSTOWN CARL STERN 02 645 3619
 CARLINGFORD ROSKO MCKAY 02 624 3353
 GLADESVILLE MARK ROTHWELL 02 817 4627
 SYDNEY EAST JACKY COCKINOS 02.344.9111
 COOMA FRED BISSELLING 0648 23263
 QLD
 BRISBANE JACK FRICKER 07 262 8869

VIC
 LATPOBE VLY GEORGE FRANCIS 051 34 5175
 VA
 KALGOORLIE TERRY BURNETT 090.21.5212

MC-10 GROUPS:
 LISMORE ROB HILLARD 066 24 3089
 SYDNEY GRAHAM POLLOCK 02 603 5028
 WARRNAMBOOL GARY FURR 055 62 7440

TANDY 1000 / MS DOS:
 NSW:
 GLADESVILLE MARK ROTHWELL 02 817 4627
 SYDNEY WEST ROGER PUTHEN 047.39.3903
 WYONG JOHN WALLACE 043 90 0312
 QLD:
 BRISBANE
 NORTH BRIAN DOUGAN 07 30 2072
 SOUTH BARRY CAWLEY 07 390 7946
 GOLD COAST GRAHAM MORPHEIT 075 51 0015

SA
 PORT LINCOLN BILL BOARDMAN 086 82 2385
 VIC:
 MELBOURNE TONY LLOYD 03 882 4664

FORTH:
 PORT LINCOLN JOHN BOARDMAN 086 82 2385
 SYDNEY JOHN REDMOND 02 85 3751

ROBOTICS:
 BOWEN TONY EVANS 077 86 2220
 GOLD COAST GRAHAM MORPHEIT 075 51 0015
 SYDNEY GEOFF FIALA 02 84 3172
 WAGGA WAGGA CES JENKINSON 069 25 2263

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

24 HOUR 300 BAUD BULLETIN BOARD SYSTEMS
 SYDNEY:
 INFOCENTRE 02 344 9511
 TANDY ACCESS 02 625 8071
 THE COCOCONNECTION 02 618 3591

HOBART:
 HUB 002 49 4 05

VIDEOTEK SYSTEMS
 MOUSETEX 059 42 5528
 VTX 4000 03 741 3295

TANDY INFO ON VIATEL
 GOLDLINK VIATEL #642#
 BLAXLAND COMPUTER SERVICES VIATEL #64263#
 COMPUTER HUT SOFTWARE VIATEL #64262#
 PARIS RADIO VIATEL #64268#
 POWER CODE VIATEL #64265#
 TANDY VIATEL #64261#

ALLAN BEALE 726953300
 FRED BISSELLING 648232630
 JACK FRICKER 726288690
 JOHN GRIGSBY 945872030
 STUART HALL 939765790
 BOB KENNY 665122050
 JEFF LARSEN 705471270
 IAIN MACLEOD 944821360
 CHRIS NAGLE 689523360
 RICHARD PANKHURST 280717870
 ROSS PRATT 648230650
 RICKEY SCHMAHL 298151500
 ARTHUR SLADE 262289400
 DARCY O'TOOLE 755105770
 RON WRIGHT 352924510

GOLDLINK

COM. STATION

642

ON

Telecom
VIATEL
AUSTRALIA'S NATIONAL VIDEOTEX SERVICE

BULLETIN BOARDS
TANDY COMPUTERS
ATARI COMPUTERS
COMMODORE COMPUTERS
FAST FLORIST
REVIEWS
HARDWARE & SOFTWARE
SHOP



SOFTGOLD MAGAZINE
Registered by Australia Post -
Publication No. QBG 4009
AUSTRALIAN CoCo
Publication No. QBG 4007
P.O. BOX 1742
SOUTHPORT, QLD, Australia, 4215.

POSTAGE
PAID
AUSTRALIA

YES WE EVEN TALK BACK !!