December 1990 Vol 3. No.6

# COCO LINK

## The Color Computer Magazine

**Featuring:**

COCO-LINK Award
Survey Results
Games and lots more

## Tandy Color Computer Hardware Sale

| | |
|---|---|
| Co-Co 2 64 K Memory [ MANUALS + JOYSTICKS] | $ 50 |
| Single Disk Drive Double Sided [CONTROLLER+MANUAL] | $ 300 |
| Twin Disk Drives Double Sided [CONTROLLER+MANUAL] | $ 400 |
| Tape Recorder CCR-83. [ 240 Volt + Cords ] | $ 35 |
| Tandy Printer DMP-106 [ With 2 extra ribbons ] | $110 |
| Serial - Parallel Printer Interface [ 75 -- 9600 Baud ] | $ 55 |
| Tandy Hi-Resolution Joystick Interface | $ 5 |

Items can be purchased separately

Contact : KEVIN GOWAN Phone : (08) 381 6740

## Tandy Color Computer Hardware Sale

| | |
|---|---|
| Co-Co 2 64 K Memory [ MANUALS + JOYSTICKS] | $ 50 |
| Single Disk Drive Double Sided [CONTROLLER+MANUAL] | $ 300 |
| Twin Disk Drives Double Sided [CONTROLLER+MANUAL] | $ 400 |
| Tape Recorder CCR-83. [ 240 Volt + Cords ] | $ 35 |
| Tandy Printer DMP-106 [ With 2 extra ribbons ] | $110 |
| Serial - Parallel Printer Interface [ 75 -- 9600 Baud ] | $ 55 |
| Tandy Hi-Resolution Joystick Interface | $ 5 |

Items can be purchased separately

Contact : KEVIN GOWAN Phone : (08) 381 6740

# Contents

## Departments

## Columns

## Features

## Advertisers

COCO LINK

EDITOR:

Robbie Dalzell

ASST. EDITOR:

Garry Holder

SUB-EDITORS:

OS9:
Ken Wagnitz

Hardware:
Darren Ramsey

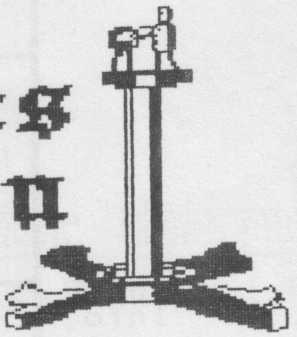
Correspondence:
Garry Holder

Submissions:
Robbie Dalzell

COCO LINK

## Copyright Notice

# Robbies Column



XMAS GREETINGS
I would like to take this opportunity on behalf of Garry and Myself to wish all our readers the very best for Xmas and the New Year. Should you be driving over the holiday period, please drive carefully. We cannot afford to lose subscribers.

******************************************************

THE END OF THE RAINBOW
It looks like the US Rainbow Magazine is going the way of other Coco orientated magazines. It has shrunk in size by a considerable amount. Rumour has it that if it gets any smaller, Falsoft will incorporate it in 'PC World', another Falsoft publication for IBM compatibles. This would be the first step to total oblivion. One wonders how long it would last.

This situation is an indication of the failing support of merchandising for the Coco. The lack of advertisers is the main cause of bringing magazines like Rainbow to their knees.

The trouble is that, with strong rumours of the Coco 3's demise after Xmas in the US, software and hardware developers are no longer willing to spend time and money developing programmes etc. for a dying market. As usual Tandy neither deny or confirm the rumour.

This does not mean that the Coco will suddenly disappear. Here in Australia we have overcome the trauma of being deserted suddenly by Tandy. The same will happen in the USA which has a much larger Coco Community.

I would suggest though, that you use the adverts, in what may be the last of your Rainbows, to get any programmes from the US that you particularly want.

******************************************************

THE MM1 - A WAY TO THE FUTURE?
The above paragraph points out a need for us to look into our future options in the computing world and start planning accordingly. Maybe the new MM1 machine just entering the marketplace will be the way to go. In the "LINK-UP" section of the magazine I have reproduced a letter from the General Manager of IMS, the makers of the MM1. It is in reply to a letter I sent regarding their plans for an Australian distribution of the machine.

I think you will find the letter interesting to say the least.

If I can obtain the necessary information from the US, I will keep COCO-LINK readers informed on the progress of this new machine. I also hope to get added help from whoever takes on the Australian end of the business. This will be done on the long term view that we must start looking to the future now.

******************************************************

COCO-LINK AWARD
This year's award goes to the NATIONAL OS9 USER GROUP for helping sustain the interest in the Color Computer at the OS9 and higher language level. Further details can be found elsewhere in this magazine.

******************************************************

BELATED CONGRATULATIONS
The news of the arrival of Johanna Vagg's new Daughter came too late for inclusion in our last magazine.

We would like to wish Johanna belated congratulations on behalf of all our readers and welcome Christina Maree into this troubled world. We wish her well and hope she finds much happiness in the future.

******************************************************

VIDEO QUIZES
Have you ever been to one of those Quiz nights down at the local club or hotel? You Know the type of thing...where you form teams and a question master asks a number of questions in sets of 10 or 20.

I have attended quite a few and though I have so far failed to win a prize, I have thoroughly enjoyed myself each time.

Jim Eadsforth of Adelaide has added a new dimension to this pastime by putting the questions on a video tape via his Coco. Jim uses his Video in conjunction with the Coco for many different purposes. He has got the technique down to a fine art.

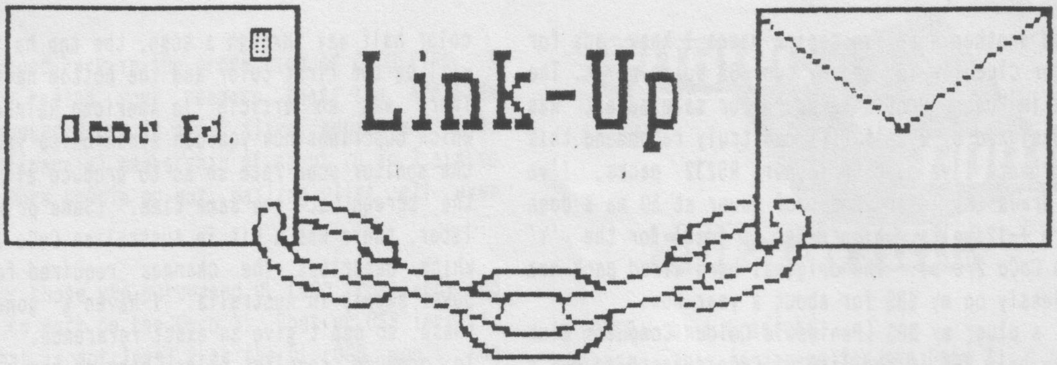
In this particular use of the video he has the questions slowly scrolling up the screen and you have to furiously put pen to paper to get the answers down. After each block of 40 questions the answers then scroll up the screen and you can see how well (or badly) you have done. There were 9 blocks of 40 questions on the tape Jim gave me. Some of the questions were visual using photographs to display the subject of the question.

The questions cover a wide variety of subjects and would be a good test for anyone.

As Jim says,"Some people don't realise that anything from the computer that is displayed on the TV screen can be transferred on to a video tape."

This surely must give some readers new ideas on what to do with their Coco. It opens the door to a wide range of

Continued on page 15



Dear Ed,
With the demise of Australian mags like Rainbow and Softgold, I have enjoyed reading COCo-LINK and will continue to do so until either I give up the CoCo world; (heaven forbid), or buy a PC of which there are several on the market worth buying.
Over the last few issues a couple of letters have appeared with regard to the size of print listings. I would like to express my comments on this particular subject. For instance, in the August copy (and others as well I suppose), the size of the print of both 'MATCHEM' and 'MENU MAKER' is a bit on the small size. Surely for the sake of an extra couple of dollars per year in subscription fees, the sheer enjoyment of typing these listings into our trusty CoCo would be that much better.
I apologise if this short note sounds like a bitch session, but I am not alone in my feelings on this matter. All in all I think your magazine is a great one.
Robert Fergusson. TAS.

This subject comes up quite often and has been answered in full in previous magazines. Suffice it to say that the reasons for doing what we are do are many and varied and genuine.

*********************************************************

Dear Ed,
Just a few lines of encouragement about your (ours too) magazine.
I have enclosed pages from 1991 American Radio Shack catalogue. As you can see the CoCo3 is still alive and well.
Also is it possible to list some of the printers Tandy use, and what to do when your software doesn't include the printer that you have in its config set up. e.g. is a DMP105 the same as a DMP106 etc.
I have a DMP106 and a DMP130a and on graphics they are like chalk and cheese; and also if I type ?MEM I get 22824 and after typing PCLEAR1 ?MEM shows only 27432 How can I get more memory to work with, because both 128K or 512K machines show the same MEM.
J. McNabb. VIC

I have reprinted a printer code chart, which appeared in a previous COCO-LIMK, elsewhere in this magazine.
Although it is not comprehensive, I hope it will be of some use to you.

The extra memory in a Coco 3 gives much more memory space for ML and other higher languages. The memory shown With MEM is the amount available for BASIC. Although this memory is the same as that of a Coco 2, it must be remembered that it all usable for programming.
Unlike the Coco 2, the 3 has separate memory allocation for text screen and hi-res graphics. This means that when programming you do not use up any of your 27432 for the text screen or graphics screen.

**********************************************************

Dear Ed,
This is a hint that may be of help to some. It saves disks and space. All disks can be used as a flip side, allowing the use of both sides.
Firstly on the left side of the disk, we have to cut out a rectangular piece the same as what already there on the other side.
It is possible to buy a cutter at most computer shops to do this job. This is all that is required to make a disk into a flip side for most computers such as Commodore etc; but more is required for the CoCo. With a sharp knife or razor blade, cut along the bottom of an old disk and remove the disk.
Place the cover over the disk you wish to make into a flip side, and with the hole on the opposite side of the present hole, mark the hole and place some form of protection under the marked hole. I use the backing paper from labels. Use a single hole punch to punch out the hole. Repeat on the other side. This makes a flip side disk, not a double sided disk.
Alan Murrells. Vic.

Thank you for this information.

**********************************************************

Dear Ed,
I have just received my first COCO-LINK magazine (October 1990), and to tell you the truth I was quite surprised. With todays spiraling prices you expect very little for your money, but when I received the magazine I found it a wealth of information, (and useful at that!). I must admit, the main reason for joining COCO-LINK was with Intertan's dropping of the CoCo, I thought I (& others) should support anything that supports the CoCo. After all, it will only be the users who will keep the CoCo alive from here on in.

I also had another surprise seeing names I knew, ads for the Computer Club I'm in, and my own BBS up in print. The RS232 pack in Eugene Hobb's hardware for sale advert was actually designed by myself. (I can truly recommend this pack, from tests I've done on other RS232 packs, I've found it draws the least amount of power at 50 ma & does NOT require +-12 volts, which makes it ideal for the 'Y' cable with CoCo 2/3's). The original hard wired pack has run faultlessly on my BBS for about a year now.

Just for a plug; my BBS (Peninsula Colour Computer Club BBS) is the only BBS in the list of CoCo supporting BBS's that is 'only' for the CoCo community and actually run on a CoCo! It has been running for about 3 years now with 3 upgrades in software, and numerous 'modifications' in between. The BBS is at present running CEBBS v2.1 and hopefully will soon be running from a hard drive.

After all that, I would like to congratulate the COCO-LINK team on their efforts, and to say 'keep up the good work'. I look forward to the next edition of COCO-LINK.

Stan Blazejewski. Vic.

For COCO-LINK to survive in the long run will require subscribers to let other Coco users know of our existence.

Your letter indicates that although members of your club receive COCO-LINK you were unaware of our existence.

I hope all subscribers will help to keep our numbers up at an economical level by telling every Coco owner they know to join COCO-LINK.

**********************************************************

Dear Ed,

Just a couple of comments relating to the October issue

Re Ken Vagnitz's comment on the DAT registers. Bit zero of Hex FF91 determines which set of DAT registers are actually used for addressing. (= zero is normal. =1 is second set) and you can switch between them by setting / resetting this bit. It allows for fast switching between memory banks where where this speed might be critical. Basic uses both sets at different times, so you can't rely on the second set retaining any particular values between separate calls to a ML routine. So for RS-DOS its only advantage is if you want to do rapid switching between memory that you can't fit in the 64K address space, like a block move of data etc.

Re the ML routine from Lindsay Bradford. I disassembled this one to have a look at it, mainly because I don't accept that there are any such absolutes with ML coding as he suggested in his text. ie that you can't change the instructions inside the loop. For those who may not have looked at the detail, I've included some alternatives that will produce a similar effect.

The relevant bit is a loop which produces rapid changes in the border color. If you change this while the monitor is doing its scan of screen memory, then the color on the screen will change at that point. ie if you change the color half way through a scan, the top half of the screen will be the first color and the bottom half the new color There was an article in American Rainbow a while back which describes how you can synchronise your program with the monitor scan rate so as to produce all 64 colors on the screen at the same time. (Same principle). A bit later, there was a bit in Australian CoCo (or CoCo Link) which describes the changes required for the 60 cycle power supply in Australia. I haven't gone looking for these, so can't give an exact reference.

To produce regular color bars on the screen, the time between each color change would have to be in some sort of sync with the monitor scan rate. Lindsay's program doesn't produce the effect described in his text on my machine (using an Amiga monitor), so I assume its scan rate is different to his monitor / TV set.

The time between color changes in the loop used is 10 memory cycles, so if you add instructions with the same elapsed time (eg 5 x NOP's or 2 x LBRN's) to double the time between color changes, then it should produce the same effect, but with the color bars twice as deep.

To demonstrate the general nature of this, I've included a variation to the program which allows you to vary the time delay between color changes, and also to have it stop scrolling after a period of time. You need this second bit if you want to use the effect in a program.

As set up, the minimum time between color changes is 22 cycles (With a 1 POKE'd to M1). You can reduce this to 17 by excluding the LEAX to break out of the loop. This time increases by 5 for each unit increase in the value POKE'd to M1. With my set up, a value of 8 in M1 produces a reasonable result. To show the timing, I've included the cycles for each instruction in the assembler code below

```
ST  LDX   #$FFFF       Max count
    CLRA               Start zero
S1  INCA         [2]   Change color
    STA   $FF9A  [5]   Do it
    LDB   #1     [2]   Min count
S2  DECB         [2]   Delay
    BNE   S2     [3]   Loop
    LEAX  -1,X   [5]   Break out
    BNE   S1     [3]   Continue
    RTS                To Basic
    END   ST
```

With this set up, you can try different values to see the effect. You can also start and stop it within a program which makes it useable for special effects. I spent a relaxing hour or so experimenting with this one, and wish to thank Lindsay for his contribution.

George McLintock. ACT.

Thank you for this information.

In a letter I received from a Canada, a COCO-LINK reader expressed the fact that he could not get Lindsay's programme to work. I assume it was a similar scan rate problem.

**********************************************************

Dear Ed,
Keep up the good work in the production of COCO-LINK. I would like to remind your readers that the Adelaide Community Computer Group is still alive, active and well. We have a financial membership of about 20 in 3 states and about 50 more peOple on our mailing list all over Australia.

A solution for those who purchased DL LOGO from Tandy and can't get it to work on the Coco 3. Boot up OS9 level 1, version 2 (that is not level 2 as the DL LOGO won't run under that). CAVE WALKER from Tandy has OS9 level 1, Ver 2.00 and will run on both the Coco 2 and 3.
Boot up CAVE WALKER and answer date prompt. When it is properly booted, take out the CAVE WALKER disk and replace it with the DL LOGO disk in drive 0. press reset and OS9 will boot again. respond to date prompt. Then on OS9: prompt, type LOGO. It works perfectly. This also works with OS9 programmes like DYNACALC, C COMPILER, and OS9 PASCAL.
Any queries can be made to me, but it should work OK. The solution was found within our group. Co-operation does work.

Laurie O'Shea SA

Thank you for the information. This seems to be information letter month at COCO-LINK.

*************************************************************************

Dear Ed,
How do you enter a ML programme into the computer. All the write-ups I have seen in the Rainbow etc., all skirt around this and do not actually say "for beginners in ML you do this"!!abcdef. The listings are there in the mags same as BASIC but they might just as well be written in Latin or Hindustani - What do you do to enter them???

Roy Simmons NSW.

To enter a ML (Assembler) listing such as that shown in a letter above, you need a 6809 Editor Assembler (EDTASM). Programmes can also be entered by putting the hex addresses in DATA statements and then POKEing them into memory using the basic READ command. The programme is then on call using the USR command.
This subject could be of interest to many readers so I will endevour to include an article on the subject in the February issue of COCO-LINK.

*************************************************************************

COCO LINK

*************************************************************************

# PUBLIC DOMAIN SOFTWARE

3 BOXES - by L.Bradford
******* This is a game for the Coco 3. I used to call it liney. It is a two player game where each tries to complete the most squares by adding one side at a time.
3 VAGAS - by A.Murrells
******* This is a three line, three reel poker machine for the Coco 3. The symbols are chess men. It is written in BASICand uses nice hi-res graphics.
ABOUT - by K.Gowan
***** Officially called 'Move About' this graphic game makes you align the 15 numbers in order on the board. It's colourful and fun.
KING TUT
******** This is a text adventure game.
MEMORY
****** A short programme to test your memory.
NAUSEA - by K.Kenny
****** This programme is based on hangman but instead of hanging the poor hero is being guilotined this time. To sve his life you must answer the questions correctly. Each wrong answer brings the blade closer to our hero's neck.
There is one snag....the questions are all on classical subjects. ie opera, history, shakespeare, mathematics and other difficult subjects. Not for the faint hearted but fun all the same.
PATIENCE - by G.McLintock
******** This game plays the cards exactly as I know the game. It uses the text screen. I found it adictive.
PONG
**** A two player game of Ping-Pong using joysticks to manipulate the bats.
PUZZLE
****** This programme works out a word puzzle and puts it to the printer. (The printer is required). This is the old puzzle where you have to find the words which are hidden in a square of determined size. The words can be horizontal, diagonal or vertical.
A great one for magazines, schools and general fun.
SLITHER
******* Hit as many targets as possible without touching the side of the screen or the end of your tail. The more targets you hit the higher the score but the snake gets longer and longer. Using the arrow keys to move the snake.
WIGWORM
******* Just like Katerpillar. Try and destroy the wiggleworm before it reaches the bottom of the screen and devours you. Joystick necessary.

*************************************************************************

**IMS** Interactive Media Systems

238 Catawba Avenue
Davidson, North Carolina 28036
704/892-6233

Dear Mr. Dalzell:

Thank you for your kind letter of August 15.

As President of Interactive Media Systems, I like to take out some time to personally answer some of our mail. I hope that the information I provide is helpful to you.

Your first question concerns availability of the MM/1 in Australia. We are currently working closely with two companies "down under" to set up dealerships. From these companies you will be able to order an MM/1 and get it delivered in a timely fashion -- and affordably! We are especially concerned about keeping the cost down for you by allowing some of the integration to occur in your country. You will also be able to buy software and hardware addons in your country.

The MM/1 is a totally new kind of computer, so you might expect that it would not have a great deal of software for it. However, it ALREADY has more software -- and of a higher quality -- than the CoCo 3 had in its first few months. For a listing of just some of the OSK professional software available, you may write to

Microware Systems Corporation
1900 NW 114th Street
Des Moines IA 50322

Ask for a copy of the Microware OS-9 Source Book. It costs $40 as I recall.

Also, the IMS Developers Association, which has over a dozen active members, is already working hard on graphics packages, animation, games, BBS, electronic mail, uucp, digital sound, MIDI, and other software.

IMS itself is also a software company. One of the first major programs from the DOS world that we are porting to the MM/1 is QuickBasic. It is due out in the first quarter of 1991 and it will open up whole new worlds to our users.

As to exactly what the computer can do, consider this. The MM/1 is designed from the ground up as a smoothly multitasking, multimedia computer. It provides DMA that won't halt the CPU while accessing floppies, hard drive, tape drive, or sound. The 1.4 Meg floppy drive that is included holds 4 times the data that a CoCo OS-9 floppy drive can hold, and it accesses tracks twice as fast. Data throughput with the floppy disk is 100% faster than the CoCo, and the hard disk performance can be 4 to 8 times faster than CoCo hard disks. With powered mice ports, joystick, bidirectional parallel ports, serial ports, MIDI and network capability, and RGB-A video, you have a computer that is second to none.

The 68070 CPU was designed by Signetics as a microcontroller for realtime computing. It is popular in factories and labs as well as in some European computer systems. It is the CPU designated for the brain of many Compact Disk-Interactive (CD-I) devices from Sony and Philips and other manufacturers.

The 68070 in the MM/1 runs at 15 MHz and includes the 100 Kbps high speed bus for networking, DMA controller, watchdog timers, and a serial port -- all on one chip! IMS has implemented the networking, allowing up to 128 MM/1s to be networked together in a neat and inexpensive way.

The MM/1 provides a built-in palette controller that lets you choose 256 colors from a palette of 16 million. This is as many as the Mac II series, and far more than the CoCo, ST, Amiga, and many VGA computers from Tandy and others.

MS-DOS emulation is still being worked on, and may not be ready this year. We hope to have something very exciting by next Spring. Our technique is similar to that used by UNIX workstations that perform MS-DOS emulation, except that the method we have chosen in much faster. However, you should be clear that, just like the UNIX workstations, we do not DESIRE to be MS-DOS compatible. Even Microsoft is phasing out MS-DOS. We do not wish to push our customers into the past.

As for support, there are at least two Australian vendors who are seeking to become MM/1 dealers. Both companies are solid and dependable. We hope to have an agreement with them by this Spring, so not only will you be able to buy your MM/1 locally, you can also get the kind of service you expect!

CoCo OS9 compatibility will be achieved through recompiling existing programs. Dozens of your favorite vendors are starting this process now. CoCo OS9 binaries will NOT run.

You will be able to run your existing Basic09 programs with little or no change, and C programs are as easy. Both languages run much faster on the MM/1, with some graphics routines running up to 80 times faster than a standard CoCo.

Graphics programs like Max 10 are being worked on now, and should beat all CoCo programs in features, speed, and value.

The MM/1 includes OS-9/68000. When you get the MM/1 for $779, you get OS-9, the C compiler, Basic, Network File Manager (for networking), PC File Manager (for reading/writing/formatting MS-DOS disks), Sequential Block File Manager (necessary when you buy a tape backup drive), a graphics editor, a text editor, demos, and more!

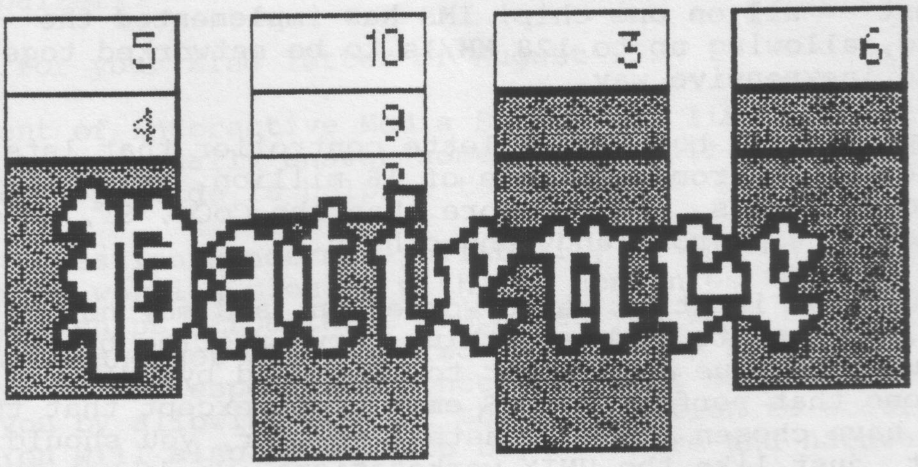
Thank you again for your letter. I have asked the corporate offices of Interactive Media Systems to send you some brochures.

Sincerely,

Paul K. Ward

Paul K. Ward

## Coco 3 (or 2) Game



### By George McIntock

INTRODUCTION
This program is written for the Coco 3 but can be easily converted back to use the 32 column screen. All that is required is to change the LOCATE commands to PRINT@ ones. The format of LOCATE is LOCATE column number , row number. Which becomes PRINT@ row number * 32 + column number. To allow for the offsets, reduce column numbers by 2. It will probably also require some minor changes to the layout, but the program itself works OK.

The listing included allows you to play the game of Patience on your computer. This is a relatively simple game to program and uses a simple display procedure on the text screen. It will only allow 'legal' moves as I understand them.
Simple card games are quite good for the purpose of following the logic of a program because you can get out a deck of cards and start playing the game itself. You then have an actual situation that you can relate directly to the program.

THE GAME:
The tableau is 28 cards in 7 piles or columns. The first pile, one card; the second, two cards; and so on up to 7 piles. The top card in each pile is face up (Exposed card), all others are face down (Concealed cards). These 'piles' are called columns in the description and program comments.
The cards remaining, after dealing the tableau, form the stock, called the Deck here. Single cards in the deck are exposed by threes. ie the next card exposed is the third one down from the 'top', or the last card. When you reach the bottom, you start from the top again. Only the 'top' card in the deck is exposed at any time.
The initial sequence of play is to build up sequences of exposed cards in the columns by moving other exposed cards from anywhere on the table (ie from the deck or other columns). Sequences are in descending values with alternating suit colors. eg red jack, black ten, red nine etc.
If moving exposed cards between columns, then all exposed cards from a column are moved as a group. When this happens, the next concealed card in the column is exposed. If there are no more concealed cards in that column, then a 'space' is created. Any exposed King, from anywhere on the table, can then be moved to that space.
When an Ace is exposed, it can be moved to start a foundation pile for that suit. This is called a pile in this descriprion, (not to be confused with the column piles). The foundation pile is built up for each suit in ascending sequence. ie Ace hearts, two of hearts, three of hearts etc. Any exposed card from anywhere on the table can be moved to a foundation pile. When moving a card from a column to the foundation pile, it must be the bottom exposed card in that column.
You win the game by getting all cards into the foundation piles. This doesn't happen very often, hence the name of patience.

REPRESENTING CARDS WITH DATA:
A normal deck of playing cards can be represented in a program in a number of different ways. In essence you have 52 separate items, each of which requires two 'tags' to uniquely identify it. eg Ace of Hearts, three of spades, nine of diamonds etc. There are 13 cards in each suit, and 4 cards of each value. If you do card games with a joker, then it normally only has one tag. If you do games like Canasta, then the wild cards may require a third tag.
The approach used here is to allocate a 'two position' number to each card which uniquely identifies the card and its suit and value at the same time. eg the first

position value identifies its suit and the second its value, or vica versa.

The data representing the cards in this program is held in strings. Most people tend to think of strings as being used only for text, possibly because this is the way they are normally used. However, strings are just another way of holding and manipulating data in a program. In the same way, 'normal' text is just a special sub set or class of data.

Strings in Basic are variable length and can be from zero to 255 bytes long. It is not necessary to specify a maximum length because string space is automatically allocated to hold the number of bytes required. Data in a string is held as a series of bytes, where each byte can have a value between zero and 255. For normal text, the value for each byte is within the range of 32 to 127.

Each card is represented by two bytes. The first byte represents its value, and the second byte its suit. The cards value is represented by its ASCII letter. eg A for ace, 2 for two, .. , 9 for nime, T for ten, J for jack etc. In concept, the second byte can be regarded as the ASCII letter for each suit. eg H for hearts, S for spades etc. So that the two byte representation becomes AH is Ace of hearts, 3C is three of clubs etc.

Line # 1550 sets the strings H$, D$ etc to equal the character value to represent each suit.

This sort of representation in strings is in fact very convenient for most card games. A full deck has 52 cards, which requires 104 bytes to represent the lot. The 108 cards used for Canasta will also fit in a single string. (216 bytes). It therefore follows that any sub group of cards in a players hand or discards etc will also fit in a single string. The size of the string identifies the number of cards in any sub group, and the position of each card in the sub group can always be retained. These aspects can simplify the logic required for many aspects of playing a game of cards on a computer.

PROGRAM LOGIC:

The GOSUB at line 30 does the initial set up

Lines 1520-1550 set up some arrays and values that are required later. The arrays contain the following data

C$(7) hold the concealed cards for each column. eg C$(1) holds the concealed cards for column #1, C$(2) for column #2 etc

E$(7) hold the exposed cards for each column.

S$(4) hold the cards for each foundation pile

CV$(5,2) holds the data required for converting the face value of a card to its numerical equivalent. eg CV$(1,1) = "A" and CV$(1,2) = "1". The code in lines 1080-1090 and 1340-1350 use this relationship to establish that an A has a numerical value of 1, a T of 10 etc

CP$(2) holds the characters which form the 'flashing cursor' on the screen during data entry. If you don't like its appearance then simply change them here. If you make these the same, then the cursor won't flash.

Other Main variables in program:

NC = Pointer to cards during the deal
Q1 = Pointer to exposed card in deck
Q$ = Exposed card in deck
CP = Pointer to cursor character
F$ = Card to move from
G$ = Card to move to
EP = Error indicator. = zero is OK

Lines 1690-1770 shuffle the cards. In practice, it is not possible to simulate a real world shuffle of a deck of cards. Mainly because there is no precise way of doing it in the real world. What we in fact do is to generate a random distribution of all the cards within a deck, and this is normally close enough. However, it is an interesting theoretical point that a computer shuffle is more likely to produce an extreme distribution in any particular hand than would occur with a normal shuffle

The method of shuffling here is fairly intuitive. You start with a full deck of cards in a natural sequence in one pile (DT$), and no cards in another pile (DK$). You select one card at random from DT$, and move it to DK$. DK$ now has one card and DT$ has 51. You continue this loop until DK$ has 51 cards and DT$ only one. You then add that single card to DK$ which now contains the 52 cards in a random sequence.

In more detail, Line 1710 generates a random number between 1 and the number of cards remaining in DT$. Line 1720 puts a copy of that card into T$. Line 1730 adds the card to the bottom of DK$. Lines 1740-1750 actually remove it from DT$. The removal from DT$ is done in two steps. T$ is set equal to the number of cards 'above' the selected card, and DT$ is then set equal to the cards 'below' it. DT$ is then 're formed' by the statement DT$=T$+DT$. In the process, the selected card is removed from the pile.

Lines 1580-1630 prints the 'fixed' bits on the screen that don't alter during the game.

Lines 1850-1900 are used for debugging and to 'find' out what is in any particular string. To find out what is in any string (hand or pile etc), Break the program, set CK$ equal to the string to examine, and GOSUB 1850. It will then be displayed on the screen

Lines 60-110 Deal the cards to each column for the initial set up for the game. The deal here follows the same natural sequence you would use when actually playing the game. The outer loop (FOR X = 1 TO 7) takes the top card from the deck and adds it to the exposed cards for each column. (E$(X)). The inner loop (FOR Y = X+1 TO 7), takes the following cards in sequence from the deck and adds them to the concealed cards for each of the rest of the columns in sequence. (C$(Y)).

Note that the cards are not actually removed from the deck string (DK$) during this process. The counter, NC, is incremented each time to 'point' to the next card in the deck. Line 120, which is executed after the deal, takes all the remaining cards from the deck (DK$) and puts them in RK$, which represents the remaining cards.

The main control loop here effectively starts at line # 130 and extends to 360. It might be noted here that the control logic for this program does not follow all the rules for a 'structured' program. These can be tedeous at times, and can in fact confuse control logic rather than simplify it. One of the big advantages of Basic is that you are not constrained to an 'artificial' structure for control logic by the program language syntax.

The general concept of control type logic used here is based on a series of inner and outer loops, which is often used for Assembler coding and can be quite effective with Basic as well.

The outer loop starts at line # 130 and performs all the operations required for the game. It starts by printing all the exposed cards in the columns on the screen. The first inner loop, Line # 140, checks if the game has been won (With the GOSUB 560). The second inner loop (Lines 150-170) prints the exposed card in the deck. The third inner loop gets the next move from the keyboard.

When you have finished processing the move, you then have to start the next move. The position in the logic where you start the next move from, will depend on what that move was. For example, if you simply expose the next card in the deck, then you don't change anything else in the game, or in the screen display, so there is no need to execute from the outer loop again after this move. You return directly to the second inner loop. You don't go back to the first inner loop either, because this move cannot win the game.

The input of data for a move (Lines 200-360) is set up to allow for the variations that can apply for this game. Most moves require two values, the current position of the card to be moved, and the position to which you want to move it. Others require only one entry, eg to expose the next card or to quit.

In this game all individual entries only require a single key press to indicate its value, so can use INKEY$ to get it. In general, INKEY$ is preferred for data entry because you don't have to press ENTER to complete a move

The data entry for a move is in two stages. First up, you get the 'move from' position into M1$. You then test this for a single value move, and perform this if it is one of these (Lines 250-270). You also check if it is a valid move entry, and if it is not, you indicate invalid and repeat this step. If it is valid, you then go on to get the second part of a normal move.

The data input here displays a 'flashing' cursor while the program waits for you to press a key (Lines 310-330). This requires a procedure to display one cursor value for a period of time and then change it at regular intervals. The method used here has a couple of components. The different cursor characters are held in an array CP$. A pointer, CP, is used to select which one will be used at any particular time. The cursor display changes according to the value of CP.

The time delay is measured by the TIMER function. For any specific machine, you can establish a regular time delay with a simple count. eg have a counter, say CN, which is increased by one each time the program executes the INKEY$ loop. When it reaches a selected value, you reset it to zero and change the cursor character.

Line 390 converts the ASCII characters from the keyboard to numbers. If either position entered is D for deck or P for pile then its numeric value would be zero. Lines 400-420 identifies any legal instances of these, and branches to the appropriate part of the program to perform the operation. If the move is from column to column, then this is done in lines 440-530.

This description could go on for ever if I tried to explain the rest of the program in the same level of detail. If you read the listing, and follow the comments, it should make sense.

For card games, or any other program as well, it is probably best to start with the layout of the game on the screen, and get this little bit to work first. You can then go on from there to add the rest.

HOW TO SUBMIT MATERIAL TO COCO-LINK
**********************************

PROGRAMMES: On tape or disk.
At least two copies should be on the tape/disk one of which should be saved in ASCII format.
Where possible include a description of your prodgramme saved as below for articles.

ML PROGRAMMES:
These require Source code saved on a suitable word processor. Two copies should be made.
A working copy of the programme should be included for checking by COCO-LINK.

ARTICLES:
At least one copy saved in ASCII format plus one copy on a commercial word processor where possible. (VIP Writer etc.)

HINTS AND TIPS:
Hand written or typed is acceptable.

LETTERS TO THE EDITOR:
Hand written letters will be accepted subject to the length. Long letters should be submitted on disk in the manner above for articles.

All disks and cassettes will be returned in due course.

# PATIENCE

```
10 'PATIENCE - The game, using Strings to represent the cards
15 CLEAR 2000
20 ' By George McLintock
25 ' MODIFIED BACK FOR COCO
27 WIDTH 40
30 GOSUB 1520 'Do setup
40 '
50 ' Deal cards from top of deck

60 NC=1:FOR X=1 TO 7
70 E$(X)=E$(X)+MID$(DK$,NC,2):NC=NC+2 'Exposed card at top of column
80 FOR Y=X+1 TO 7
90 IF Y=8 THEN 110
100 C$(Y)=C$(Y)+MID$(DK$,NC,2):NC=NC+2 'Concealed cards for each column
110 NEXT Y,X
120 RK$=MID$(DK$,NC): Q1=5 'Rest of cards to Deck & First exposed in deck
130 GOSUB 1410 'Print Cards
140 GOSUB 560: IF WIN > 0 THEN CLS:LOCATE 12,12: PRINT "YOU WIN": STOP
150 IF Q1 >= LEN(RK$) THEN IF Q2=0 THEN Q2=1: Q1=LEN(RK$)-1 ELSE Q2=0: Q1=5: GOTO 150 'Roll over at end of deck
160 Q$=MID$(RK$,Q1,2) 'Extract it
170 LOCATE 21.16: PRINT "D=";Q$; 'Print it
180 '
190 'Get next move from keyboard

200 LOCATE 3,15: PRINT "MOVE FROM ";STRING$(12," ");:CP=1:T = TIMER+40
210 LOCATE 14,15: PRINT CP$(CP);:M1$=INKEY$:IF M1$ <> "" THEN 240
215 M1$=INKEY$: IF M1$ <> "" THEN 240
220 IF TIMER > T THEN T = TIMER + 40: IF CP=1 THEN CP=2 ELSE CP=1 'Flash Cursor
230 GOTO 215
240 LOCATE 14,15: PRINT M1$;:IF M1$="N" THEN Q1=Q1+6: GOTO 150 'Do next in deck
250 IF M1$="Q" THEN RUN 'Give up and re start
260 IF M1$="D" THEN 300
270 IF M1$ = "?" THEN 1210
280 IF M1$ >= "1" AND M1$ <= "7" THEN 300
290 GOSUB 1160:GOTO 200 'Entry is invalid
300 LOCATE 16,15: PRINT "TO";:CP=1:T = TIMER
310 LOCATE 20,15: PRINT CP$(CP);:M2$=INKEY$:IF M2$ <> "" THEN 340
315 M2$=INKEY$: IF M2$ <> "" THEN 340
320 IF TIMER > T THEN T = TIMER + 40: IF CP=1 THEN CP=2 ELSE CP=1 'Flash Cursor
330 GOTO 315
340 LOCATE 20,15: PRINT M2$;:IF M2$="P" THEN 390
350 IF M2$ >= "1" AND M2$ <= "7" THEN 390
360 GOSUB 1160:GOTO 300 'Entry invalid
370 '
380 'Intrepret input and do it
390 E1=VAL(M1$):E2=VAL(M2$) 'Move is from E1 to E2
400 IF E1=0 AND E2=0 THEN 600 'Deck to pile
410 IF E1=0 THEN 670 'Deck to column
420 IF E2=0 THEN 740 'Column to pile
430 ' Is from column E1 to E2
440 IF LEN(E$(E2))=0 THEN F$=LEFT$(E$(E1),2): GOSUB 980: GOTO 460 'Is space
450 G$=RIGHT$(E$(E2),2):F$=LEFT$(E$(E1),2):GOSUB 1020
460 IF EP <> 0 THEN 200 'INVALID

470 E$(E2)=E$(E2)+E$(E1)
480 'Expose next concealed card in column
490 IF LEN(C$(E1))=0 THEN E$(E1)="":GOTO 530
500 IF LEN(C$(E1))=2 THEN E$(E1)=C$(E1):C$(E1)="":GOTO 530
510 E$(E1)=RIGHT$(C$(E1),2)
520 C$(E1)=LEFT$(C$(E1),LEN(C$(E1))-2)
530 GOTO 130
540 '
550 'Check for win
560 IF LEFT$(S$(1),1)="K" AND LEFT$(S$(2),1)="K" AND LEFT$(S$(3),1)="K" AND LEFT$(S$(4),1)="K" THEN WIN = 1
570 RETURN
580 '
590 'Move from deck to pile
600 G2$=RIGHT$(Q$,1):GOSUB 910
610 F$=Q$:G$=S$(S):GOSUB 1290
620 IF EP <> 0 THEN 200 'Invalid move
630 S$(S)=Q$:GOSUB 830
640 GOSUB 1480: GOTO 160
650 '
660 'From deck to Column E2
670 IF LEN(E$(E2))=0 THEN F$=Q$:GOSUB 980: GOTO 690 'Column empty
680 G$=RIGHT$(E$(E2),2):F$=Q$:GOSUB 1020
690 IF EP <> 0 THEN 200 'Invalid

700 E$(E2)=E$(E2)+Q$:GOSUB 830
710 GOTO 130
720 '
730 'From Column E1 to pile
740 F$=RIGHT$(E$(E1),2):G2$=RIGHT$(F$,1):GOSUB 910
750 G$=S$(S):GOSUB 1290
760 IF EP <> 0 THEN 200 'Invalid
770 S$(S)=F$
780 IF LEN(E$(E1))=2 THEN 490 'Last in column
790 E$(E1)=LEFT$(E$(E1),LEN(E$(E1))-2)
800 GOTO 130
810 '
820 'Remove from deck string
830 IF LEN(RK$)=2 THEN RK$="--":RETURN
840 IF Q1=1 THEN RK$=MID$(RK$,3):RETURN 'Is top card
850 T$=LEFT$(RK$,Q1-1)
860 RK$=T$+MID$(RK$,Q1+2)
870 Q1=Q1-2
880 RETURN
890 '
900 'Find suit - for pile
910 IF G2$=S$ THEN S=1
920 IF G2$=H$ THEN S=2
930 IF G2$=D$ THEN S=3
940 IF G2$=C$ THEN S=4
950 RETURN
960 '
970 'Check top card is king
980 EP=0: IF LEFT$(F$,1) <> "K" THEN 1160
990 RETURN
1000 '
1010 'Check validity of move to column
1020 ' F$ = Card to move from : G$ = Card to move to
1030 F1$=LEFT$(F$,1):F2$=RIGHT$(
F$,1)
1040 G1$=LEFT$(G$,1):G2$=RIGHT$(
G$,1)
1050 IF F2$=H$ OR F2$=D$ THEN IF
 G2$=H$ OR G2$=D$ THEN 1160
1060 IF F2$=S$ OR F2$=C$ THEN IF
 G2$=S$ OR G2$=C$ THEN 1160
1070 FOR X=1 TO 5
1080 IF F1$=CV$(X,1) THEN F1$=CV
$(X,2) 'Convert from A T J Q K
1090 IF G1$=CV$(X,1) THEN G1$=CV
$(X,2) 'To numeric equivalent
1100 NEXT X
1110 F=VAL(F1$):G=VAL(G1$)
1120 IF G-F <> 1 THEN 1160
1130 EP=0:RETURN
1140 '
1150 'ERROR ON INPUT
1160 SOUND 10,10
1170 SOUND 32,10
1180 EP=1:RETURN
1190 '
1200 'Display number of conceale
d cards
1210 CLS: LOCATE 3,0:PRINT "Numb
er of concealed cards in each co
lumn"
1220 FOR X=1 TO 7: LOCATE 3,X:PR
INT "Column";X;"=";INT(LEN(C$(X)
) / 2);"CARDS"
1230 NEXT X: LOCATE 3,X+1: PRINT
 "Number of cards in deck =";INT
(LEN(RK$) / 2)
1240 PRINT: INPUT "Press ENTER t
o continue ";T
1250 GOSUB 1580: GOTO 130
1260 '
1270 'Check validity of move to
pile
1280 ' F$ = Card to move from :
G$ = Card to move to
1290 F1$=LEFT$(F$,1):F2$=RIGHT$(
F$,1)
1300 G1$=LEFT$(G$,1):G2$=RIGHT$(
G$,1)
1310 IF G1$="-" THEN IF F1$="A"
THEN 1390 ELSE 1160 'Pile is emp
ty
1320 IF F2$ <> G2$ THEN 1160
1330 FOR X=1 TO 5
1340 IF F1$=CV$(X,1) THEN F1$=CV
$(X,2) 'Convert from A T J Q K
1350 IF G1$=CV$(X,1) THEN G1$=CV
$(X,2) 'To numeric equivalent
1360 NEXT X
1370 F=VAL(F1$):G=VAL(G1$)
1380 IF F-G <> 1 THEN 1160
1390 EP=0:RETURN
```

```
1400 '
1410 'Print exposed cards
1420 FOR X=1 TO 7
1430 RX=1:CX=3*X+1:Y=1
1440 FOR Z=1 TO 13
1450 IF Y > LEN(E$(X)) THEN P$="
 " ELSE P$=MID$(E$(X),Y,2)
1460 LOCATE CX,RX: Y=Y+2:PRINT P
$;
1470 RX=RX+1:NEXT Z,X
1480 RX=11: CX=26: FOR X=1 TO 4
1490 LOCATE CX,RX: PRINT "P ";S$
(X);:RX=RX+1:NEXT X
1500 RETURN
1510 '
1520 DIM C$(7),E$(7),S$(4),CV$(5
,2),D$(4)
1530 CV$(1,1)="A":CV$(1,2)="1":C
V$(2,1)="T":CV$(2,2)="10":CV$(3,
1)="J":CV$(3,2)="11":CV$(4,1)="Q
":CV$(4,2)="12":CV$(5,1)="K":CV$
(5,2)="13"
1540 CP$(1)="?":CP$(2)="="
1550 S$="S":H$="H":D$="D":C$="C"
'Suits
1560 CLS:PRINT"SETTING UP": X=RN
D(-TIMER)
1570 GOSUB 1640 'Set up deck
1580 CLS:Y=4:FOR X=1 TO 7
1590 LOCATE Y,0:PRINT X;:Y=Y+3
1600 NEXT X
1610 LOCATE 3,18: PRINT "Use  N
to expose next card in deck": L
OCATE 3,19:PRINT "     Q  to Qui
t"
1620 LOCATE 3,20:PRINT "    OR
? to display number of
concealed cards"
1630 RETURN
1640 DATA A,2,3,4,5,6,7,8,9,T,J,
Q,K
1650 DT$="":S$(1)=S$:S$(2)=H$:S$
(3)=D$:S$(4)=C$
1660 FOR Y=1 TO 4:A$=S$(Y): GOSU
B 1790: S$(Y)="-"+S$(Y): NEXT Y
'Set up initial deck
1670 '
1680 'Shuffle Deck
1690 DK$="":FOR X=1 TO 52
1700 IF LEN(DT$)=2 THEN DK$=DK$+
DT$:GOTO 1760
1710 C=RND(LEN(DT$)/2) 'Select c
ard at random
1720 T$=MID$(DT$,(C-1)*2+1,2) '
Extract that card
1730 DK$=DK$+T$ 'Add it to botto
m of DK$
1740 T$=LEFT$(DT$,(C-1)*2):DT$=R
```

```
IGHT$(DT$,LEN(DT$)-LEN(T$)-2)
1750 DT$=T$+DT$ 'Remove it from
DT$
1760 NEXT X
1770 RETURN
1780 'Add 13 cards for suit in A
$
1790 FOR X=1 TO 13
1800 READ T$:DT$=DT$+T$+A$
1810 NEXT X
1820 RESTORE:RETURN
1830 '
1840 'Display a String of cards
in CK$
1850 T=1:Q=INT(LEN(CK$)/2): IF Q
=0 THEN RETURN
1860 FOR Y = 1 TO Q
1870 PRINT MID$(CK$,T,2);" ";
1880 T=T+2:NEXT Y
1890 PRINT
1900 RETURN
```

# Eagle Poker

## By Mark Funston & Shaun Butterley

This is a basic Poker machine game written for the Coco 3. It was written as a High school project on communications by the two authors.

The game is straight forward in its operation and should hold no difficulty for the user. Have fun!!!

```
0 :
15 POKE&HFFD9,0:REM HI-SPEED POK
E
20 ' ****      pokie       ****

30 '         BY:
40 '      MARK FUNSTON
50 '           &
60 '     SHAUN BUTTERLEY
70 ' A RED'N'FUNGI PRODUCTION
80 WIDTH40:PRINT"              I
NSTRUCTIONS:":PRINT:PRINT:PRINT"
  TO INSERT A COIN PRESS THE SP
ACEBAR.":PRINT:PRINT" YOUR TOTAL
 AMOUNT OF COINS TO BEGIN    WIT
H IS 50, ALTHOUGH YOU CAN CHANGE
 IT  BY THE VALUE OF TC IN LINE
220"
90 PRINT"YOU WILL BE TOLD WHEN Y
OU RUN OUT OF     COINS. YOUR TOT
AL COINS LEFT ARE IN THE TOP LEF
T CORNER OF THE SCREEN. "
100 PRINT:PRINT:PRINT"        GO
OD  LUCK!!!"
110 POKE&HE6B7,&H39:HSCREEN2:POK
E&HE6B7,&H20:HCLS4:PALETTE0,0:HC
OLOR0,4:RGB
120 FOR B=1TO6:HBUFF B,1253:NEXT
B
130 FOR C=1TO6
140 ON C GOSUB 1110,1220,1330,14
40,1550,1660
200 HGET(0,0)-(41,55),C
210 HCLS 4:NEXT C
220 TC=50
230 HCOLOR0:HLINE(103,69)-(230,1
26),PSET,B
240 PALETTE2,7:PALETTE3,56
250 HCOLOR14:HPRINT(35,23),"R'N'
F":HCOLOR0
260 HLINE(91,43)-(244,191),PSET,
B
270 HLINE(101,67)-(232,128),PSET
,B
280 HLINE(122,151)-(217,173),PSE
T,B
290 HLINE(110,43)-(113,31),PSET:
HLINE-(224,31),PSET:HLINE-(227,4
3),PSET
300 HLINE(244,103)-(252,108),PSE
T:HLINE-(252,118),PSET:HLINE-(25
2,118),PSET:HLINE-(244,124),PSET

310 HLINE(114,173)-(225,181),PSE
T,B:HLINE(114,173)-(122,167),PSE
T:HLINE(225,173)-(217,167),PSET
320 HPAINT(92,44),3,0:HPAINT(115
,32),9,0
330 HPRINT(16,4),"VIDEO SLOTS"
340 HLINE(230,43)-(223,55),PSET,
B:HLINE(226,45)-(226,53),PSET
350 HPRINT(27,7),"20c"
360 HPAINT(224,45),3,0:HPAINT(10
2,68),3,0
370 HPAINT(246,108),3,0
380 HPAINT(115,175),3,0:HPAINT(2
20,171),3,0:HPAINT(120,171),2,0
390 HCIRCLE(124,151),10,0,2.2,0,
.25
```

```
400 HPAINT(125,152),2,0
410 HCIRCLE(215,151),10,0,2.2,.2
5,.5
420 HPAINT(214,152),3,0
430 HLINE(246,113)-(252,60),PSET
:HLINE(250,113)-(254,60),PSET:HC
IRCLE(253,53),7,0:HPAINT(253,53)
,14,0
440 HPAINT(253,61),0,0
450 HLINE(246,113)-(250,113),PSE
T:HPAINT(247,112),0,0
460 HCOLOR0
470 HLINE(0,180)-(32,186),PSET,B
:HPAINT(1,181),10,0:HLINE(30,186
)-(28,192),PSET:HPAINT(1,189),10
,0
480 HLINE(0,180)-(11,158),PSET:H
LINE-(10,180),PSET
490 HLINE(6,170)-(0,159),PSET:HL
INE-(8,145),PSET:HLINE-(8,162),P
SET
500 HLINE(5,150)-(0,140),PSET:HL
INE-(8,124),PSET:HLINE-(6,148),P
SET
510 HLINE(4,130)-(0,122),PSET:HL
INE-(7,107),PSET:HLINE-(4,129),P
SET
520 HLINE(4,112)-(0,103),PSET:HL
INE-(6,94),PSET:HLINE-(3,110),PS
ET
530 HLINE(3,98)-(0,91),PSET
540 HLINE(4,95)-(4,76),PSET
550 HCOLOR0:HPRINT(0,0),"COINS L
EFT:"
560 PALETTE1,49:HPAINT(9,179),1,
0:HPAINT(0,168),1,0:HPAINT(2,159
),1,0:HPAINT(0,150),1,0:HPAINT(2
,140),1,0:HPAINT(0,130),1,0:HPAI
NT(2,122),1,0:HPAINT(0,112),1,0:
HPAINT(2,102),1,0:HPAINT(0,97),1
,0
570 HCOLOR0:HCIRCLE(20,76),16,0,
.8,.5,.9
580 HCIRCLE(20,73),18,0,.8,.5,.9

590 HLINE(3,73)-(0,52),PSET
600 HLINE(34,65)-(60,138),PSET:H
LINE-(30,68),PSET
610 HPAINT(0,88),13,0
620 HCOLOR13:HLINE(34,65)-(49,66
),PSET:HLINE-(36,70),PSET:HLINE-
(56,70),PSET:HLINE-(37,73),PSET:
HLINE-(59,74),PSET:HLINE-(38,77)
,PSET
630 HLINE-(50,78),PSET:HLINE-(39
,79),PSET:HLINE-(65,84),PSET:HLI
NE-(40,81),PSET:HLINE-(66,89),PS
ET:HLINE-(41,87),PSET:HLINE-(69,
98),PSET:HLINE-(44,92),PSET:HLIN
E-(72,101),PSET
640 HLINE-(44,96),PSET:HLINE-(72
,108),PSET:HLINE-(46,100),PSET:H
LINE-(74,113),PSET:HLINE-(48,104
),PSET:HLINE-(73,119),PSET:HLINE
-(48,109),PSET:HLINE-(73,125),PS
ET:HLINE-(49,111),PSET:HLINE-(67
,133),PSET:HLINE-(51,114),PSET
650 HLINE(30,68)-(18,81),PSET:HL
INE-(32,73),PSET:HLINE-(19,91),P
SET:HLINE-(34,78),PSET:HLINE-(22
,99),PSET:HLINE-(37,84),PSET:HLI
NE-(25,105),PSET:HLINE-(39,90),P
SET:HLINE-(29,109),PSET:HLINE-(4
1,97),PSET:HLINE-(29,117),PSET:H
LINE-(44,102),PSET
660 HLINE-(33,120),PSET:HLINE-(4
5,107),PSET:HLINE-(35,134),PSET:
HLINE-(47,112),PSET:HLINE-(42,13
8),PSET:HLINE-(49,113),PSET:HLIN
E-(50,138),PSET
661 PRINT:PRINT:PRINT:PRINT"PRES
S <ENTER> TO BEGIN":EXEC44539
663 PALETTE0,0
665 POKE&HE6E4,&HE6:HSCREEN2:POK
E&HE6E4,&HE7
670 HCOLOR4:HLINE(88,4)-(248,24)
,PSET,BF:HLINE(0,8)-(32,24),PSET
,BF:HCOLOR0:HPRINT(0,1),TC
680 M=RND(6):N=RND(6):P=RND(6):H
PUT (104,70)-(145,125),M,PSET
690 EXEC 43345:GOSUB 760
700 HPUT (146,70)-(187,125),N,PS
ET
710 EXEC43345:GOSUB760
720 HPUT (188,70)-(229,125),P,PS
ET
730 EXEC43345:GOSUB760
740 A=A+1:IF A=10 THEN 780
750 GOTO680
760 Y=RND(40)
770 RETURN
780 SOUND255,5
790 SOUND150,1
800 GOSUB860:IF TC=0 THEN HSCREE
N0:CMP:CLS:PRINT"SORRY GAMBLER,
I ATE ALL YOUR   MONEY. YOU ARE
STONE BROKE!!!!!":FORD=1TO5000:N
EXTD:GOTO1750:
810 HCOLOR15:HPRINT(15,17),"PRES
S":HPRINT(20,17)," BUTTON":HPRIN
T(14,18),"TO INSERT COIN"
830 PALETTE15,0
840 A$=INKEY$:IFA$<>CHR$(32)THEN
840 ELSEA=0:TC=TC-1:GOTO670
850 POKE&HFFD8,0:END
860 REM CHECK TO SEE IF WON
870 IFM=1ANDN=1ANDP=1THEN890ELSE
IFM=2ANDN=2ANDP=2THEN970ELSEIFM=
3ANDN=3ANDP=3THEN1010ELSEIFM=4AN
DN=4ANDP=4THEN1030ELSEIFM=5ANDN=
5ANDP=5THEN1050ELSEIFM=6ANDN=6AN
DP=6THEN1070ELSEIFM=3ANDN=3THEN9
50ELSEIFM=6ANDN=6THEN990ELSE880
880 IFM=1ANDN=1THEN910ELSEIFM=2A
NDN=2THEN920ELSEIFM=4ANDN=4THEN9
30ELSEIFM=5ANDN=5THEN940ELSERETU
RN
890 WIN=10:GOSUB1085:RETURN
910 WIN=4:GOSUB1085:RETURN
920 WIN=5:GOSUB1085:RETURN
930 WIN=20:GOSUB1085:RETURN
940 WIN=2:GOSUB1085:RETURN
950 WIN=50:GOSUB1085
955 GOSUB1820:RETURN
970 WIN=14:GOSUB1085:RETURN
990 WIN=25:GOSUB1085:RETURN
1010 WIN=200:GOSUB1085
1015 FOR CYC=1TO3:GOSUB1820:NEXT
 CYC:RETURN
1030 WIN=100:GOSUB1085
1035 FOR CYC=1TO2:GOSUB1820:NEXT
 CYC:RETURN
1050 WIN=5:GOSUB1085:RETURN
1070 WIN=100:GOSUB1085
1075 FOR CYC=1TO2:GOSUB1820:NEXT
 CYC:RETURN
1085 HPRINT(12,1),"YOU WIN ":HPR
INT(20,1),WIN:HPRINT(24,1),"COIN
S":TC=TC+WIN:RETURN
1090 RETURN
1100 END
1110 REM ORANGE
1120 PALETTE5,38:PALETTE6,52
1130 HCOLOR6:HCIRCLE(21,28),18,6

1140 HPAINT(21,28),5,6
1150 RETURN
1220 REM PLUM
1230 PALETTE7,32:PALETTE6,33
1240 HCIRCLE(21,28),16,7
1250 HPAINT(21,28),6,7
1260 RETURN
1330 REM SEVEN
1340 HCOLOR7:HLINE(4,6)-(37,6),P
SET:HLINE-(37,16),PSET:HLINE-(22
,46),PSET:HLINE-(10,46),PSET:HLI
NE-(25,16),PSET:HLINE-(4,16),PSE
T:HLINE-(4,6),PSET
1350 HPAINT(5,7),14,7
1360 RETURN
1440 REM MELON
1450 PALETTE13,20:PALETTE14,36
1460 HCIRCLE(21,23),18,13,.80,0,
.5
```

```
1470 HCIRCLE(21,23),17,13,.8,0,.
5
1480 HCIRCLE(21,23),16,13,.8,0,.
5
1490 HCOLOR13:HLINE(3,23)-(36,23
),PSET
1500 HPAINT(21,24),14,13
1510 HCOLOR14:HLINE(3,23)-(36,23
),PSET
1520 HCOLOR0:HLINE(12,25)-(12,26
),PSET:HLINE(15,30)-(15,29),PSET
:HLINE(20,27)-(20,26),PSET:HLINE
(30,27)-(30,28),PSET:HLINE(18,25
)-(18,26),PSET
1530 HLINE(25,26)-(25,25),PSET
1540 RETURN
1550 REM CHERRY
1560 HCIRCLE(22,32),7,14:HCIRCLE
(9,30),7,14:HPAINT(22,32),14,14:
HPAINT(9,30),14,14
1570 HCIRCLE(34,25),13,13,1.4,.5
,.75
1580 HCIRCLE(34,23),24,13,.6,.5,
.75
1590 RETURN
1660 REM BAR
1670 PALETTE9,54:PALETTE10,34
1680 HCOLOR10
1690 HLINE(3,20)-(37,34),PSET,B:
HLINE(7,23)-(32,31),PSET,B
1700 HPAINT(4,22),9,10:HPAINT(8,
25),9,10
1710 HPRINT(1,3),"BAR"
1720 HLINE(3,20)-(7,23),PSET:HLI
NE(37,34)-(32,31),PSET:HLINE(37,
20)-(32,23),PSET:HLINE(7,31)-(3,
34),PSET
1730 RETURN
1740 END
1750 CMP:CLS:PRINT"     A RED'N'
FUNGI CREATION":PRINT"        FO
R THE COCO III":PRINT"
     FROM":PRINT:PRINT"        E
AGLE SOFTWARE":END
1760 '***************************

1770 '!   ALL ' REM STATEMENTS !

1780 '!   MAY BE OMITTED FOR   !

1790 '!    PRACTICAL REASONS   !

1800 '***************************


1810 ' SOUND FOR WIN
1820 SOUND 50,1:SOUND100,1:SOUND
150,1:SOUND200,1:RETURN
```

## ROBBIE'S COLUMN

Continued from page 2

ideas where your imagination is the only limitation. The educational uses of this particular quiz idea could be far reaching.

I am sure Jim would be happy to share his knowledge of this technique should anyone be interested.

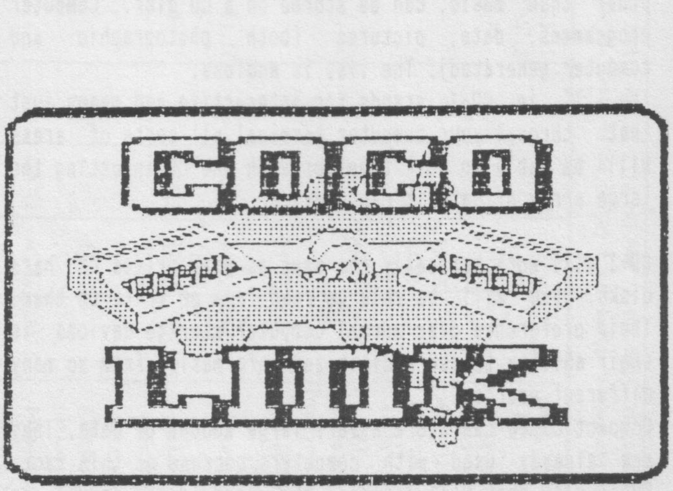
Contact: Jim Eadsforth
Ph. (08) 298 2843

*******************************************************

RUMOURS
I will use this space to scotch the rumour that APD is about to cease dealing in Color Computer Hardware and software. John Poxon of APD told me personally that this is not fact and that as long as Coco owners are looking for Coco material, APD will supply it.
This rumour has become fairly widespread in South Australia. Again, I repeat, it is not true and anyone who spreads tells you it is is doing both APD and the Coco community in general a disfavour.

*******************************************************

Merry Xmas
& Happy New Year
Robbie & Garry



# CD-I

## The Multi-media Connection

By Robbie Dalzell

The 1990's promise to be the CD-I (Compact disk - Interactive) years. This new technology is hailed as the step which will take computers into the multi-media arena in a big way and into the twenty-first century.

In a multi-media conferance in London last June, the directors of Sony Corporation and Mitsushita Electric Industries gave a commitment to work together to achieve hardware and software compatibility standards for this new technology

What is meant by Multi-media?
Basically it means to use various mediums to get a message across. This can mean interfacing data from an independant source along with photographs, comuter graphics, music, voice etc. etc. You name it and it will be possible.

What does CD-I stand for?
We all know what a CD player is. It is a 'computer' which plays our compact disk music recordings. Most people look on the CD player as an electrical appliance but in reality it is a highly sophisticated computer for reading digitally encoded instructions from a disk. Many things, other than music, can be stored on a CD disk...Computer programmes, data, pictures (both photographic and computer generated). The list is endless.
The "I" in CD-I stands for interactive and means just that. through your computer terminal all sorts of areas will be able to work together with the CD inputting the large array of data it can provide.

CD-I will work basically the same as disk drives or hard disks. you will be able to read from or write to them. Their preference over normal computer storage devices is their ability to store digitized information from so many different sources.
Compact disks can store a very large amount of data. They are already used with computers because of this fact. Eventually they will replace the hard disk drive. An example of the vast storage capacity is that the entire Encyclopedia Britannica has been put on to five CD disks. This includes animated sequences and interactive maps. These are available now at about $200.00 each. You would have to own the special equipment to use them, of course.

The professional market for CD-I is developing fast. Many firms realise its potential for training. For instance, the French car maker, Renault, has implemented a CD-I system for dealer training that provides detailed information about marketing their cars. One CD-I disk contains all the information in several languages so that dealers around the world can use the same disk and choose their native language.

The home market offers unlimited potential for CD-I in the consumer electronics market. To start with we will probably see interactive games, childrens educational/ entertainment and reference manuals.
You could have the complete maintainance manual for your car on one CD disk. You could even have 5 or 6 of these enormous books on how to use DBASE III or Lotus 123 on the one disk. Wouldn't that be exciting!

However as CD-I takes root new and imaginative uses will be seen, such as living books, interactive adventures and truly elaborate educational material.

CD-I is certain to be a winner in the consumer market. Like the CD Audio machines CD-I will be designed to provide:

Easy access
Simplicity of use.
High quality.
Good visual design.
and one standard to be used worldwide.

The 1980's saw the CD player bloom. The 1990's will do the same for CD-I. Sometime in the not too distant future you may be able to dial up a central computer and get a full home control, burglar alarm system or information on any subject you wish to name.
Many companies are working on these concepts now along with many, many other weird and wonderful ideas.

Like most inovative and new things, CD-I will be very expensive to start with, but like its cousin the CD player, as it gains in popularity prices will tumble. We can look forward to affordable CD-I systems that will do things unimaginable at the moment.

END

# Printer Codes

| Print Function | DMF-200 | DMF-130 | DMF-120 | DMF-100 | DMF-105 | Sakala | Epson | Gemini | Amust | Brother | Riteman | Super 5 | OMP110 | DWP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Backspace | 8 | 8,n112 | | | 8 | | | | | 8 | 8 | 8 | 8,n | |
| Bold, End | 27,32 | 27,32 | | 30 | 27,32 | 27,14 | 27,87,0 | 27,87,0 | 27,70 | | 27,70 | 27,70 | 27,32 | 27,32 |
| Bold, Start | 27,31 | 27,31 | | 31 | 27,31 | 27,14 | 27,81,1 | 27,87,1 | 27,69 | 27,69 | 27,69 | 27,69 | 27,31 | 27,31 |
| Compressed, 12cpi | 27,23 | 27,23 | | | 27,23 | | | | | 27,77 | 27,77 | 27,77 . | 27,23 | 27,14 |
| Condensed, 17cpi | 27,20 | 27,20 | 27,20 | | 27,20 | 27,15 | 27,15 | 27,15 | 15 | 15 | 27,15 | 15 | 27,20 | |
| Correspon. Compress | | 27,29 | | | | | | | | | | 27,119 | 27,29 | 27,15 |
| Correspon. Normal | 27,19 | 27,18 | | | | | | | | | | | 27,18 | |
| CR | | 27,22 | | 26 | | | | | | 13 | | | 26 | |
| CR + LF | 13 | 13 | 13/141 | 10/13 | 13 | 27,13 | 27,13 | 27,13 | 27,13 | 14 | 13 | 13 | 13 | |
| Elongation, End | 27,15 | 27,15 | 27,15 | | 27,15 | | 27 87 0 | | 27,87,0 | 27,87,0 | 27,72 | 27,87,0 | 27,15 . | |
| Elongation, Start | 27,14 | 27,14 | 27,14 | 31 | 27,14 | | 27 87 1 | | 27,87,1 | 27,87,1 | 27,71 | 27,87,1 | 27,14 | |
| Form Feed | 27,52,n | | | | 12 | 27,12 | 27,12 | 27,12 | 27,12 | 12 | 12 | 12 | 12 | |
| Graphics, End | | 30 . | 30 | | | | | | | | | | | |
| Graphics, Start | | 18 | 18 | 18 | 18 | | | | | | | | 18 | |
| Italics, End | | 27,56,0 | | | | 27,116 | 27,53 | 27,53 | 27,82,10 | 27,53 | 27,53 | 27,53 | | |
| Italics, Start | | 27,55,1 | | | | 27,115 | 27,52 | 27,52 | 27,82,19 | 27,52 | 27,52 | 27,52 | 27,66 | |
| Justify, Right | | 27,17 | | | | | | | | | 27,81 | | | |
| LF, full forward | 27,54 | 27,54 | 27,54 | | 27,54 | | 27,10 | 27,10 | 27,10 | 10 | | 10 | 27,54 | |
| LF, full reverse | 27,10 | 27,10 | | | | | ,106,14 | | | | | | | |
| LF, half forward | 27,29 | 27,29 | 27,28 | | 27,28 | | | | | | | | 27.28 | |
| LF, half reverse | 27,30 | 27,30 | | | | | | | | | | | | |
| LF, n/24 forward | | 27,64,n | | | | | | | | | | | | |
| LF, 1/12 forward | 27,50 | 27,50 | 27,50 | | | | | | | | | | | |
| LF, 1/24 forward | | 27,57 | | | | | | | | | | | | |
| LF, 1/36 forward | | 27,51 | | | | | | | | | | | | |
| LF, 1/6 forward | | | | | | 27,50 | 27,2 | 27,50 | 27,51 | 27,50 | 27,50 | 27,50 | 27,54 | |
| LF, 1/8 forward | | 27,26 | | | | 27,48 | 27,1 | 27,48 | 27,48 | 27,49 | 27,48 | 27,48 | | |
| LF, 3/4 forward | 27,55 | 27,56 | 27,56 | | 27,56 | | | | | | | | | |
| Microfont | | 27,77 | | | | | | | | | | | 27.77 | |
| Normal, 10cpi | 27,19 | 27,19 | 27,19 | | 27,19 | 27,77 | 27,77 | 27,53 | | | 27,80 | 27,80 | | |
| Page Length | | | | | | | 27,67,n | 27,67,n | | 27,67,n | 27,67,n | | | |
| Prop. Spaced | 27,17 | 27,17 | | | | 27,112, | 27,112, | | 27,112, | 27,112,n | 27,112 | | 27,17 | |
| Sel. International | | | | | | | | | | 27,82,n | 27,82 | | | |
| Skip perforation | | | | | | | | | | 27,78,n | | 27,78,n | | |
| Subscript, End | | 27,89 | | | | | 27,84 | 27,84 | | 27,84 | 27,84 | | | |
| Subscript, Start | | 27,83,1 | | | | 27,83,1 | 27,83,0 | 27,83,1 | 27,83,1 | 27,83,1 | 27,83 | | 27,28 | |
| Superscript, End | | 27,89 | | | | | 27,84 | 27,84 | | 27,84 | 27,84 | 27,84 | | |
| Superscript, Start | | 27,83,0 | | | | 27,83,0 | 27,83,0 | 27,83,0 | 27,83,0 | 27,83,0 | 27,83 | 27,83 | 27,30 | |
| Underline, End | 14 | 14 | 14 | 14 | | | 27,45,0 | 27,45,0 | 27,45,0 | 27,45,0 | 27,45 | 27,45,0 | 14 | 14 |
| Underline, Start | 15 | 15 | 15 | 15 | | | 27,45,1 | 27,45,1 | 27,45,1 | 27,45,1 | 27,45 | 27,45,1 | 15 | 15 |
| Unidirect., End | | | | | 27,85,0 | | 27,85,0 | 27,85,0 | | 27,87,0 | 27,85 | | | |
| Unidirect., Start | | | | | 27,85,1 | | 87,85,1 | 27,85,1 | | 27,85,1 | 27,85 | 27,85 | | |

As you may notice, there are a lot of holes above. As some printers use the same code, maybe one can swap one code for another. Then again, you can fill in the holes.

Epsom COMPATIBLES

| | |
|---|---|
| NATIONAL PANASONIC | 1080<br>1091 |
| OKIDAKA | 190<br>290 |

# Color Computer 3

# Interfaced with Commodore Monitor

## By Kevin Gowan

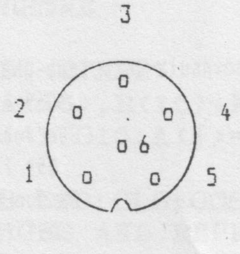
It is possible to use the 1084S-P Commodore monitor with the Coco 3. This utilises the RGB (Analogue) output of the Color Computer (10 pin header on underside) with the 6 pin 'Lin.RGB' DIN socket in the rear of the 1084 monitor.

NOTE: The 1084S-P Commodore monitor is needed - DO NOT USE the 1084S-D version, as the 6 pin DIN socket is not provided (a single 'D' type socket is supplied instead).

You are provided with a variety of interconnecting cords with the 1084S-P monitor (to suit various computer types), but unfortunately, none of them is suitable for use with the Coco 3; you will need to have a special cord made to suit.

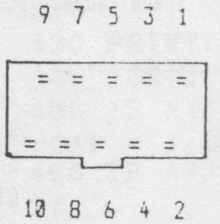
Wiring details:-

| 6 p/plug | Function | 10 p/header | Pin# CM8 Signal |
|---|---|---|---|
| | | | 1 - Not connected |
| 3 | Ground | 1,2 | 2 - Ground |
| 4 | RED \ VIDEO | 3 | 3 - RED \ VIDEO |
| 1 | GREEN > VIDEO | 4 | 4 - GREEN > VIDEO |
| 5 | BLUE / VIDEO | 5 | 5 - BLUE > VIDEO |
| | Missing-Pin | 6 | * - ---------- |
| | Sound O/P | 7 | 7 - Sound O/P |
| 2 | Horiz Sync | 8 | 8 - H-Sync |
| 6 | Vert Sync | 9 | 9 - V-Sync |
| | | 10 | 10 - Not Connected |

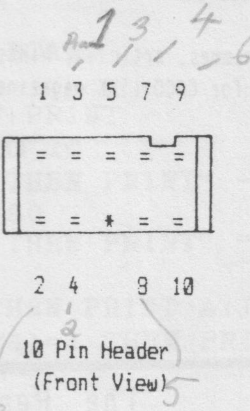


6 Pin DIN plug
(Rear View)
(Wiring Side)
[1084S Monitor]

1-Brown
2-Blue.
3-orange
4-yellow
5-green.
6-purple



10 Pin Header
(Rear View)
(Wiring Side)
[Color Computer 3]



10 Pin Header
(Front View)
[CM8 MONITOR]

8,7

Note: Pin 6 of the 10 pin header socket (underside of Coco 3) is missing, as it is not used. When using the 10 pin plug, remove pin 6, and replace it with a sliver of plastic dipped in super glue, to effectively block off the 6th pin position, thus making the header plug polarised - i.e. It will only be able to be connected one way to the Coco 3.

It is possible to use a 10 wire ribbon cable to connect to the 10 pin header plug. The other end may be split to extract the wires connected to pins 2 and 7, which can be connected to a RCA plug for the sound (ensure the wire from pin 2 connects to the outer skirt of the RCA plug, as it is at earth potential). The wires from pins 1, 3, 4, 5, 8,and 9 can then be connected to the 6 pin DIN plug at pins 3, 4, 1, 5, 2 and 6 respectively - refer to wiring chart above.

You may use a 6 wire cable for the video signals in leiu of the ribbon, and you could also utilise the audio RCA socket on the rear of the COCO 3 using an audio RCA to RCA cable for the sound.

# Club Noticeboard

## CoCo Supporting BBS's

| | |
|---|---|
| Penninsula CoCo club BBS | Ph. (03)-580-4605 |
| Happy Hacking BBS | Ph. (03)-787-8759 |
| Hard Rock Cafe BBS | Ph. (03)-894-2815 |
| Real Connection #1 | Ph. (03)-808-0810 |
| Real Connection #2 | Ph. (03)-808-0331 |
| Decedence BBS | Ph. (03)-794-7949 |
| Nemisis BBS | Ph. (03)-331-1155 |
| J&M Systems BBS | Ph. (02)-749-1935 |

All support 2400 baud and 8 bit word length, No parity, one stop bit.

## WANTED URGENTLY

Programmes, articles, hints and tips for COCO-LINK magazine.

## NATIONAL OS9 USER GROUP

The fullest OS9 information service in Australia.

Monthly magazine included in annual subscription of $18.00

Write now to:

Gordon Bentzen
8 Odin St.
Sunnybank
Qld. 4109

## CLUB CONTACTS

| | |
|---|---|
| Adelaide.........Laurie O'Shea | 08 363 2647<br>(after 7.30pm) |
| . | Glenys Ferres<br>08 332 4264 |
| Basic............Johanna Vagg | 068 522 943 |
| Brisbane North....M.Webster | 07 285 6551 |
| Brisbane S/W......Bob Devries | 07 372 7816 |
| Geelong...........Alan Murrells | 052 753 065 |
| Moe User Group....Joseph Hester | 051 277 817 |
| . | Ian Taffs<br>051 275 751 |
| OS9 User Group....Gordon Bentzen | 07 354 5141 |
| Peninsula CCC...Bob Charlesworth | 059 791 922 |
| . | Sue Jones<br>059 773 292 |
| . | Gordon Chase<br>059 711 553 |

The Peninsula Color Computer Club (Inc) meets on the 3rd Wednesday of every month - Except December - at the Mechanics Institute Hall at Frankston from 7pm to 11pm. Fee for non-members is $3 per night.

Contact:

| | |
|---|---|
| Bob Charleston | 059 791922 |
| Sue Jones | 059 773292 |
| Gordon Chase | 059 711553 |
| PCCC-BBS 2130-0700 | 03 5804605 |

Open days are held on the last Sat. of the school holidays from 12 noon - 5pm. Please call one of the above to confirm dates.

ALL WELCOME

**Coco 1,2&3** **Game**

# Two Basic Games

## By Tim Hartnell

The following two short programmes are games in their most basic form. Both have merit in that they are easily adapted and improved.

NUMBER PUZZLE
-------------

This basic game uses the text screen. The game consists of a rectangle with 15 squares which contain the numbers 1 to 14 and one blank. On the command, the numbers are shuffled round. The idea is then, by moving one number at a time, to reset the numbers to the original set pattern. This game is well known by all kids from 9 to 90.

For those with a penchant to make something more of it, the code lends itself to all sorts of ideas for change. It could be changed to a hi-res graphics screen for instance and have some colour added. Try it!

```
5 'PUZZLE PROGRAM
10 ' THE AYRTON PUZZLE
20 CLS:'HOME
30 '
40 A=RND(-TIMER)
50 DIM A(16),B(16):M=0
60 FOR J=1TO16:A(J)=J:NEXT J:GOS
UB 420
70 GOSUB 520:CLS:GOSUB420:'REARR
ANGE PIECES AND REPRINT
90 GOSUB150:'ACCEPT MOVE
100 GOSUB420:'PRINT OUT BOARD
110 GOSUB260:'CHECK FOR WIN
129 IF WINFALL = 1 THEN PRINT"YO
U COMPLETED IT IN";M;"MOVES!";EN
D
130 GOTO90
150 'ACCEPT MOVE
160 M=M+1:PRINT:PRINT"THIS IS MO
VE NUMBER";M
170 INPUT"PLEASE ENTER NUMBER OF
 PIECE TO MOVE";G
180 IF G=0 THEN END
190 J = 1
200 IF A(J)=G THEN230
210 IF J<16 THEN J=J+1:GOTO 200
220 PRINT"INVALID MOVE":GOTO170
230 A(GAP)=A(J):A(J)=16
240 RETURN
260 'CHECK FOR WIN
270 WINFLAG =0:K=0
280 FOR H = 1 TO 16
290 IF A(H) = H THEN K=K+1:B(K)=
H
300 NEXT H
310 IF K=16 THEN WINFLAG =1
320 IF X=0 THEN 390
330 PRINT:PRINT"YOU HAVE";K;"IN
THE RIGHT":PRINT"POSITION";
340 FOR N = 1 TO K
350 IF B(N)<16 THEN PRINT B(N);
360 IF B(N)=16 THEN PRINT" --";
370 NEXT N
380 PRINT
390 RETURN
400 END
410 'PRINT OUT BOARD
420 '
430 PRINT:PRINT:PRINT
~~440 FOR J = 1 TO 16~~
450 IF A(J)=16 THEN PRINT" -- ";
:GAP = J:GOTO 480
460 IF A(J)<10 THEN PRINT" ";A(J
);
470 IF A(J)>9 THEN PRINT A(J);
480 IF 4*INT(J/4)=J THEN PRINT
490 NEXT J
500 RETURN
510 'REARRANGE PIECES
520 '
530 PRINT:INPUT"PRESS A KEY TO R
EARRANGE A BOARD";J$
540 FOR J = 1 TO 16:A(J)=0:NEXT
J
550 FOR J = 1 TO 16
560 N=INT((RND(1000)/1000)*(J+2)
)+1:IF N>16 THEN560
570 IF A(N)<>0 THEN 560
580 A(N)=J
590 NEXT J
600 RETURN
610 'CHECK IF POSSIBLE
```

```
620 '
630 CK=0
640 FOR J= 1 TO 16
650 N=A(J)
660 FOR H=J+1 TO 16
670 IF A(H)=16 THEN 690
680 IF A(H)<N THEN CK=CK+1
690 NEXT H
700 NEXT J
710 PRINT ,CK;"INVERSIONS"
720 IF INT(CK/2)<>CK/2 THEN PRIN
T ,"POSITION IS IMPOSSIBLE":GOTO
540
730 RETURN
1000 FORX=1TO10:PRINTRND(1):NEXT
```

WORDBILD
--------

This is another BASIC game which lends itself to many suggestions for change.

The game can be played by 1 to 10 players. The computer provides a number of letters and the player makes up as many words as possible from them. His/her score accumulates as each word is added. When finished it passes on to the next player. The score is recorded for each player and each rounds' score is added to the total. At the end of the game the winner is announced.

The code is in its most basic form and so many things can be done to make the game much more enjoyable. For instance:

1. A timer could be added so as each player had a limited time to pick his words.
2. The programme could be made to dismiss words of less than 4 letters.
3. The programme does not have a dictionary so correction of the words must be left to the players themselves. Incorrect words can not be removed once they are typed in and (enter)ed. Putting a word deletion system with the accompanying loss of score would be a great improvement to the game.

There are many ways to make a game such as this fit individual needs. It's educational value makes it a worthwhile project.

Let us see some improved versions of the above games!

```
 10 'WORD BUILDER
 20 'BY TIM HARTNELL
 30 'ADJUSTED FOR COCO BY GARRY H
OLDER
 40 GOSUB300:'INITIALISE
 50 '************************
 60 FOR R = 1 TO 5:FOR J = 1 TO N
 70 WT = 0:M$="":FOR K = 1 TO 10:
 B$(K)= A$(INT((RND(1000)/1000)*
 37 + 1)):M$=M$ + " "+ B$(K):NEXT
  K
80 CLS:PRINT"Round number";R;"fo
r ";N$(J)
90 FOR Z = 1 TO WT:PRINT W$(Z),:
NEXT Z:PRINT
100 PRINT"Current score is ";S(J
):PRINT"Your letters are:":PRINT
M$
110 PRINT:PRINT"Type in your wor
d, then press   the ENTER key."
120 PRINT"Type in 99 when you ha
ve run out of words."
130 E$="":INPUT C$:Q=1
140 IF C$= "99" OR C$= "" THEN 2
40
150 D$=MID$(C$,Q,1):IF ASC(D$)>9
0 THEN D$=CHR$(ASC(D$)-32)
160 FLAG=0:FOR W = 1 TO 20
170 IF MID$(M$,W,1)=D$ THEN FLAG
=1
180 NEXT W:E$ = E$ +D$
190 IF FLAG=0 THEN PRINT D$;"is
not allowed":FOR X = 1 TO 999:NE
XT X:GOTO80
200 IF Q< LEN(C$) THEN Q = Q+1:G
OTO 150
210 H= LEN(C$):S(J) = INT(S(J) +
H^3)
220 WT = WT+1:W$(WT)=E$
230 PRINT:GOTO80
240 PRINT:PRINT:PRINT
250 NEXT J:PRINT:PRINT:PRINT:NEX
T R
260 PRINT:PRINT"the final scores
 are:"
270 FOR J = 1 TO N:PRINT TAB(8);
N$(J);" :";S(J):NEXT J
280 END
290 '*************************
300 'initialise
310 CLS
320 DIM A$(37),B$(10),N$(10),S(1
0),W$(200)
330 FOR J=1 TO 26:A$(J)=CHR$(64+
J):NEXT
340 FOR J=27 TO 37:READ A$(J):NE
XT J
350 FOR J=1 TO 37:PRINT A$(J);"
";:NEXT
360 PRINT:PRINT"    Welcome to wo
rd builder":PRINT
370 INPUT"how many players (1 TO
 10)";N:N=INT(N):IF N<1 OR N>10
THEN370
380 PRINT:FOR J=1 TO N:PRINT"Ent
er name of player number";J:INPU
T N$(J):NEXT J
390 RETURN
400 '************************
410 DATA "E","E","E","A","E","I"
,"O","U","A","O","U"
```

# Readers Survey Results

Normally the results of a survey such as this are solely for the use of the magazine staff to help formulate the future direction of the magazine. However, one of the suggestions submitted was that the results of the survey should be published. Here then are the results

Firstly, the total number of surveys returned accounts for only 25% of our total subscriptions. I deem this enough to count it as a fair cross section of our readership. The results are mainly expressed in percentages.

I will take each survey question in turn and comment where I think it necessary.

1) Our subscribers are predominantly male with an average age of between 36 and 65

2) a) We have only 11% exclusively cassette users.
   b) Over 60% use disk BASIC exclusively. Many use both disk and cassette.
   c) 10% of subscribers use OS9 Level1 10% of the time.
   d) 33% use Level 2 up to 40% of the time with only 2% using it exclusively.

3) This section was a revelation showing that, in the main, the interests of our subscribers are spread fairly evenly across the whole spectrum of subjects. The percentages were worked out on the total number of points accumulated using the 0-5 numbering scale. Graphics was surprisingly the most popular subject.

| | | | | | |
|---|---|---|---|---|---|
| Amateur radio | 15 | BASIC 09 | 31 | Beginner info | 52 |
| Business | 26 | C Language | 21 | Desktop Pub. | 44 |
| Education | 41 | Games | 52 | Graphics | 92 |
| Hardware | 46 | Home finance | 22 | Home help | 34 |
| ASSEMBLY | 45 | OS9 Lev 1&2 | 45 | Reviews | 54 |
| Sound & music | 42 | Technical | 61 | Telecomm | 38 |
| Tutorials | 61 | Utilities | 68 | | |

| | | | |
|---|---|---|---|
| Robbie's Column | 75 | Better BASIC | 65 |
| Graphics by | 45 | Link-up | 69 |
| Chain reaction | 53 | | |

5) LISTINGS:

| | | | |
|---|---|---|---|
| a) Type in: | Frequently 58 | Sometimes 48 | Never 24 |
| b) Length : | Too long 10 | Too short 10 | Right 80 |
| c) Amount : | Too many 0 | Too few 51 | Right 37 |

As can be seen, many of our readers feel that there are too few listings in COCO-LINK. It must be stressed that nearly all listings published in the magazine are submitted by you, the subscriber. We do not receive enough submissions to increase this ratio. At present, 90% af all material submitted is submitted by a small core of dedicated people. To them we should all be thankful. Without them COCO-LINK will cease to exist.

6) DO YOU PROGRAMME?

| | | | | | |
|---|---|---|---|---|---|
| BASIC | 94 | BASIC 09 | 9 | FORTRAN | 2 |
| C | 11 | COBOL | 0 | ASSEMBLY | 31 |

This is a percentage of those who programme in each language. It must be taken into consideration that a considerable number of readers programme in more than one language whilst others do not programme at all. This was a question I did not ask, so I have no exact figure of the percentage of non-programmers.

7) 85% of the subscribers returning the form said that they were willing to submit material to COCO-LINK. This is one of the most important questions on the form. As explained earlier, COCO-LINK depends on your submissions to carry on. If you have shown a willingness to submit material, please do it now. Don't put it off until it is too late!

8) Everyone seems to get their COCO-LINK in good condition. If there is any problems with the new packaging please let us know.

There was also a space left for your suggestions and comments. Many of you made use of this space to congratulate us on doing a good job. For this we thank you. It is heartening to be told that all the time we put into turning out the magazine is not wasted.

There were quite a few suggestions which we thought had merit. I found that most of these depended more on the magazine subscribers than the editors. Some suggestions have been implemented already while others will wait on reader input.

Here is a rundown of some of your suggestions with comments added:

1. Have a list of BBS numbers for modem users.

This was implemented on the noticeboard page of the October Issue.

2. Encourage interstate news.

We are quite willing to print any news of the happenings of clubs etc. Use us to communicate.

3. Publish the results of the survey.
This is it.
4. More short programmes.
You send them in and we will publish them.
5. Run competitions.
We have run three competitions since the inception of the magazine. In all 3 cases we had only one entrant.
6. Print names and towns of subscribers.
It is not the policy of COCO-LINK to issue lists of subscribers for a variety of reasons.
7. More tuition on OS9.
I think the "Beginners Diary" goes a bit of the way in this direction. Any other OS9 tutorial material would have to be submitted by one of our readers with the necessary knowledge. Any takers?
10. Amateur Radio programmes.
Again, this depends on reader input. As the survey has shown there is some interest in both amateur radio and telecommunications.
11. Public Domain for Cassette users.
We have no plans to institute this service. As explained in a previous issue of COCO-LINK, we neither have the equipment to do a good job or the necessary time to do it.
12. Stick address label on outside of packaging for better viewing.
Done.

As you can see, where possible we have already instigated a number of the suggestions received. Others are up to you.

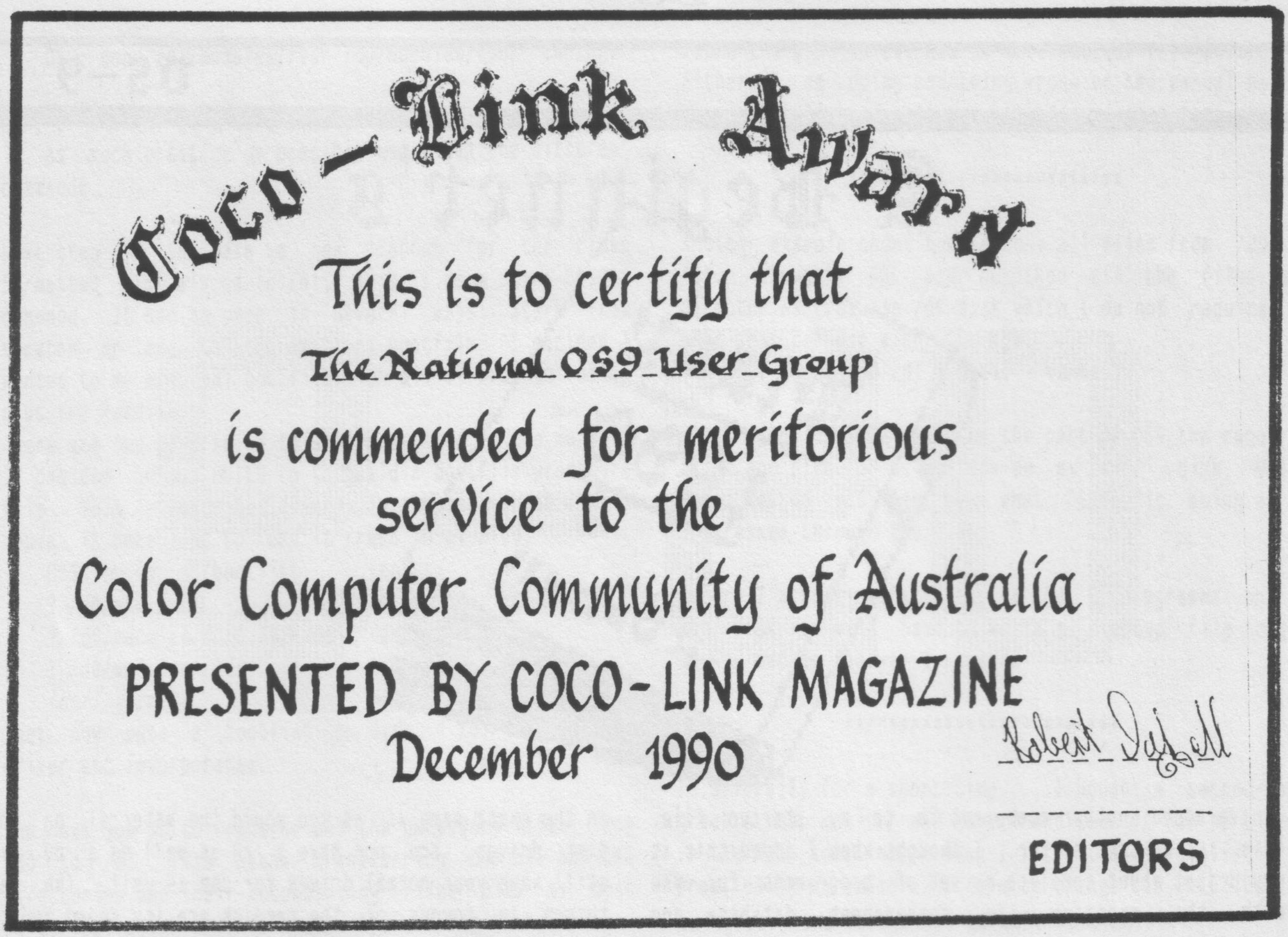
Thanks to all of you who took the time to fill in the survey and return it to us. I am sure you will agree that it shows that, basically, we are filling most needs.

**Utility**

# Dirsort

This short listing will sort the programmes in a disk directory into alphabetical order. I fin it very useful as it makes it easier to find individual programmes on a crowded disk.

```
2 PMODE0:PCLEAR1
3 CLEAR8000
4 DIM D1$(72),D2$(72):Z$=STRING$
(32,255)
5 CLS:PRINT:PRINTTAB(4)"director
y pack and sort":PRINT:PRINT:PRI
NT
6 PRINT" THIS UTILITY WILL PACK
AND SORT THE DIRECTORY ON YOUR D
ISKS.     THIS WILL MAKE YOUR DIR
ECTOR-   IES LOOK BETTER AND MAK
E FILES  EASIER TO FIND.":FORX=1
TO1000:NEXT
7 GOSUB40
8 CLS:PRINT"  INSERT THE DISK TO
BE PACKED      AND SORTED INTO DRI
VE."
9 PRINT:PRINTTAB(9)"PRESS [enter
]"
10 GOSUB43
11 PRINT:PRINTTAB(6)"reading dir
ectory..."
12 FORS=3 TO 11:DSKI$ A,17,S,A$,
B$
13 SB=(S-3)*8:D1$(SB)=LEFT$(A$,3
2)
14 D1$(SB+1)=MID$(A$,33,32)
15 D1$(SB+2)=MID$(A$,65,32)
16 D1$(SB+3)=MID$(A$,97)
17 D1$(SB+4)=LEFT$(B$,32)
18 D1$(SB+5)=MID$(B$,33,32)
19 D1$(SB+6)=MID$(B$,65,32)
20 D1$(SB+7)=MID$(B$,97)
21 NEXT
22 FORI=0TO71:D2$(I)=Z$:NEXT
23 PRINT:PRINTTAB(11)"packing...
"
24 I=0
25 FORJ=0TO71: IF ASC(D1$(J))=0 T
HEN27
26 D2$(I)=D1$(J):I=I+1
27 NEXT:L=I-1
28 PRINT:PRINTTAB(11)"sorting...
"
29 FORI=0 TO L-1:FORJ=I TO L
30 IF D2$(I)>D2$(J) THEN T$=D2$(
I):D2$(I)=D2$(J):D2$(J)=T$
31 NEXT:NEXT
32 PRINT:PRINTTAB(6)"writing dir
ectory..."
33 FORS=3TO11:SB=(S-3)*8
34 A$=D2$(SB)+D2$(SB+1)+D2$(SB+2
)+D2$(SB+3)
35 B$=D2$(SB+4)+D2$(SB+5)+D2$(SB
+6)+D2$(SB+7)
36 DSKO$ A,17,S,A$,B$
37 NEXT
38 PRINT:PRINT:PRINTTAB(7)"proce
ss complete."
39 END
40 CLS:PRINT:INPUT"WHICH DIRECTO
RY DO YOU WISH TO  PACK AND SORT
 (0,1,2,3) ";A
41 IFA<0 OR A>3THEN40
42 PRINT@484,"press [enter] to c
ontinue";
43 IF INKEY$<>CHR$(13) THEN43
44 RETURN
```



Coco-Link Award

This is to certify that

The National OS9 User Group

is commended for meritorious service to the

Color Computer Community of Australia

PRESENTED BY COCO-LINK MAGAZINE

December 1990

Robert Dalzell

EDITORS

The COCO-LINK AWARD is presented annually to individuals or organisations who the editors of COCO-LINK consider have contributed something of value to the Coco Community in Australia. $100.00 is awarded each year.

This year's COCO-LINK AWARD goes to the NATIONAL OS9 USER GROUP for organising and maintaining this very important section of the Australian Coco world.

I think we all owe a debt of gratitude to Gordon Bentzen, Bob Devries and Don Berrie for taking up the challenge and bringing new life to the OS9 User Group after it had fallen by the wayside. These three gentlemen deserve all the accolades we can bestow on them.

The monthly newsletter of the National OS9 User group carries information and programming for the budding, as well as the more advanced OS9 user. The Group also maintains a large library of OS9 Public Domain software which is available to subscribers at minimal cost. The three office holders named above have always been found willing to help straighten out the problems of the less informed.

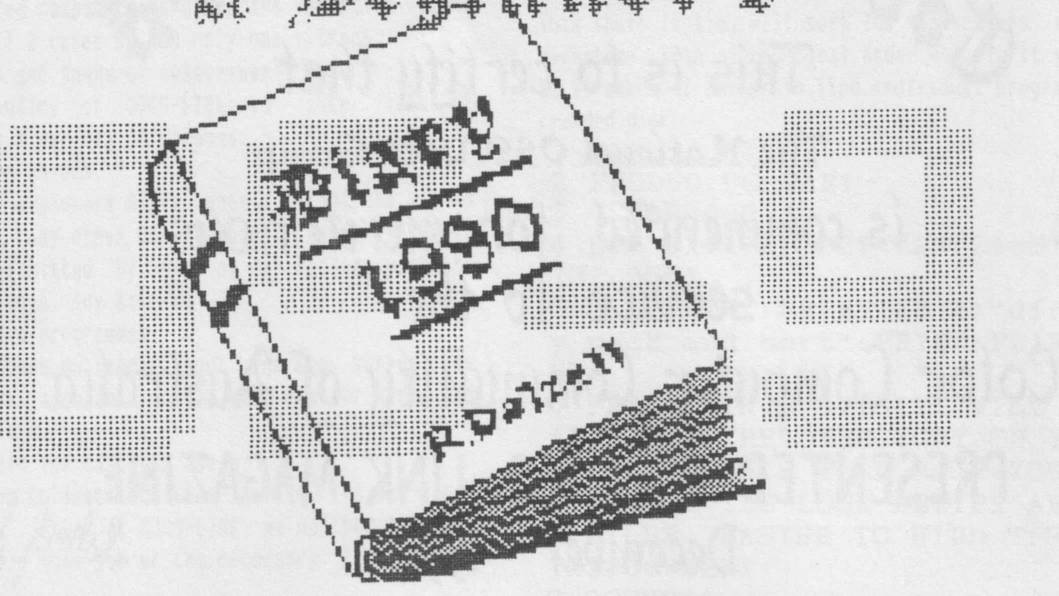
In presenting the $100.00 cheque to the NATIONAL OS9 USER GROUP we hope, in some small way, to help ease the financial burdon of running a national group such as this and, in so doing, insure the continuance of the group.

We heartily congratulate you on a job well done.

Robbie Dalzell & Garry Holder<br>For COCO-LINK Magazine.

(Details on how to subscribe to the NATIONAL OS9 USER GROUP can be found in the Noticeboard page in this magazine).

# A Beginner's



I have added another component to to my startup file. This is a text editor. I thought when I added this it would just about complete my set of programmes for use with the magazine (ie spreadsheet, database and wordprocessor.

I made one miscalculation. The text editor is not a word processor. It is a SCREEN text editor. I find I will need a text formatter to go with the text editor to format my pages of print for sending to the printer. I have found a suitable one in an old Rainbow magazine. The only drawback is that it is written in "C".

The whole purpose of this exercise is to learn to programme in "C". So, I think maybe I have got far enough with OS9 to move on to the next step in the exercise .....that is to start setting up a disk purely for use with the "C" language.

*************************

I have been told that using the "C" language directly from the disk is a very slow way to go about it. "C" is a very disk orientated language. I have it from an authority on the subject that the best way to run "C" is is on a RAMdisk as this avoids the continual disk usage.

What do I know about RAMdisks?....Not very much. So let's look into this subject before going any further. I am lucky that I have a Ramdisk programme for OS9 on one of the OS9 public domain disks I have procured. (These are available from the OS9 User Group).

Basically a Ramdisk is configured by using a part of the memory in the computer and telling it that it is a disk drive. (Computers are stupid things, you can tell them just about anything and they will believe you). You can then load files or programmes on to this Ramdisk just as you would a normal disk drive. The material is accessed in the exact same way as you would the material on your disk drives. You now have a /r0 as well as a /d0. You still have your normal drives for use as well. The main things in fovour of the ramdisk are its speed and the fact that it saves a lot of wear and tear on your disk drives, especially when used with a programme like "C".

*************************

Ok. This sounds like good advice, so how do I go about it.

This Ramdisk (we call it /r0) should be in the bootfile. This saves time energy and memory. The Ramdisk I have has two files.....ddr0.192K.dd and r0dr, that is, the device descriptor and the device driver. (A device is a data source or destination such as a window, disk drive, printer or, in this case, a ramdisk etc., etc.)

Setting up this system requires making up a new disk with a new bootfile incorporating the above. These files do not appear in the CONFIG programme so we will have to make up a new boot in some other manner.

With advice and a read through the manual, OS9GEN is the way this can be done. Firstly, the necessary descriptors and drivers must be in the MODULES directory so as OS9GEN can get at them. So, my first job is to transfer these two files from my public domain disk in /d1 to the modules directory in /d0. As I am going to use my original bootfile with the addition of the Ramdisk, I found it easier to make up a new MODULES directory on that disk to contain the two Ramdisk files. This was accomplished with:

```
   OS9: makdir MODULES
```

With the directory in place I then transferred the two files as follows:

```
   OS9: copy /d1/modules/ddr0.192k.dd /d0/modules
        /ddr0.192k.dd    <enter>
```

```
    OS9: copy /d1/modules/r0dr /d0/modules/r0dr  <enter>
```

Many of the things I do are unnecessary but I like to get in as much practice as possible using all the different commands.

Next step is to make a new OS9boot for the clean formatted disk placed in /d1. This is done using OS9GEN command. It can be used in several ways. Start from scratch or add to the existing bootfile. I decided I wanted to my original bootfile with all it had to offer plus the Ramdisk.
There are two ways to do this as described in the manual. I decided to use BUILD to to build a bootfile procedure file. This is described clearly in the manual but I will repeat it here just to keep it fresh in my mind.

```
    OS9: build /d0/bootlist          <enter>
    ? /d0/os9boot                    <enter>
    ? /d0/modules/ddr0.192k.dd       <enter>
    ? /d0/modules/r0dr               <enter>
    <ctrl><break>
```

That now puts a bootlist on disk in /d1 with the ram driver etc incorporated.

The next job is to transfer all the necessary files from /d0 to /d1. This means transferring the SYS and CMD directories and their accompanying files plus my text editor,ds, which will be needed for working with "C".
OK. How do I go about transferring these directories? I certainly do not want to transfer them file by file.

*************************

Again, by reading through the Commands section of the manual I find that DSAVE is the command needed for transferring whole directories. So let's try it! I have my system disk in /d0 and my new "C" disk in /d1
In the examples shown on page 6-41 it shows how to DSAVE to a procedure file "makecopy" and then transfer this to the required disk. I will try this method.

*************************

I have tried this several times and each time I come up with an error. I LISTed the "makecopy" file and found that it makes a procedure file to transfer ALL the files in the current data directory. I was only trying to transfer the CMDS directory. I didn't realise when I read the manual that when it said "one or more directories" that it meant complete data directories. You learn something new every minute.
I thought I had to MAKDIR the directory you are directing the copies to but I was wrong again. The procedure file does this unless you include the option -m to tell it not to makdir any directories. This was the first error I got.
After I had DELDIRed the CMDS directory on /d1 I tried again but to no avail. I then tried several ways to get round the problem but had no more success than before. Either I am doing something wrong or the manual has an error. I will have to get a bit of advice on that one.

*************************

Another example shows how to copy all files from /d0 to /d1. I have DEL and DELDIRed all the files and directories from the /d0 disk which I do not require on this disk for use with "C". Now:

```
    OS9: dsave /d0 /d1 ! shell   <enter>
```

After considerable work on the part of OS9 the required files and directories are now on my #1"C" disk. Using this method you can see what DSAVE is doing as it progresses through the disk.

Before I actually start loading the "C" programme on to the disk I will have to write a "startup" file to get things set up the way I require.

*************************

To diversify for a short time.....I bought a second-hand green screen monitor and have found that it can be a pain in the neck sometimes. When I fire it up initially it is practically unreadable. I have to change the PALETTEs for background and foreground to make it elegible.
The same problem was found in OS9. When I booted up the initial screen was hard to read, especially as I have it set up for 80 columns. I thought I would have to re-CONFIG a new bootfile from scratch to put suitable back and foregrounds. Up till now I have INIZed a window with a shell so as I could see what I was doing.

While looking for some other information in the manual I came across MODPATCH. I have used this command before but being new to the game it had slipped my memory. Anyway, you can alter any device using this command without having to go back to step 1. Here is all that is required to alter the colours of your screen. You call up MODPATCH and then alter the terminal (TERM) like this:

```
    OS9: modpatch         <enter>
    l term
    c 33 2 2
    c 34 3 0
    c 35 3 2
    v
    <ctr><break>
then:
    OS9: cobbler /d0
```

This gave me a light coloured screen with black letters and a black border.
Look up modpatch and then you can do a bit of experimenting. For colour televisions you can make the screen any colours you wish.

*************************

# Data Master

## An Overview and tutorial

### By Armand Belanger

DATA MASTER is a data base programme which is very practicable, useful and user friendly. It also has a superb means of correction at a finger tip. However, anyone who buys the programme and who tries to use it without help and depending only on the instruction manual, has a good chance of failing to properly utilise the programme. To make things easier for learners there is a tutorial to guide you to success.

Fistly, the disk you obtain on buying the programme cannot not be used directly because its 'dir'is equipped with only DM. (Short for DATA MASTER). You must add it to the "Commands Directory" of OS9 to be called on booting. If you know of a programmer in a club, ask him to do it for you. Otherwise, place OS9 master disk in drive0 and your DATA MASTER disk in drive1 then, after you boot the OS9 system disk, type the following each time the OS9: prompt appears after you boot the OS9 Level II master disk.

```
OS9:chd /d1                     <enter>
OS9:copy /d1/dm/d0/cmds/dm      <enter>
OS9:w80                         <enter>
OS9:montype r (with RGB)        <enter>
OS9:                            <CLEAR>
OS9:dm                          <enter>
```

and voila, you have DATA MASTER on screen. Type / and use the right arrow key to see what happens. You can now remove the DATA MASTER disk as you no longer need it. DM is now on the OS9 master disk. The next time you use DM, it will be enough to boot the master disk and type:

```
OS9:w80                         <enter>
OS9:montype r                   <enter>
OS9:                            <CLEAR>
OS9:dm                          <enter>
```

CREATING A FILE

Say you wish to create a list of names, addresses and telephone numbers like the following in 132 columns, printed in condensed print. (You could use less characters if you wish).

| Surname | first name | Address | Town | State/country | Postcode | Telephone |
|---|---|---|---|---|---|---|
| Dalzell | Robert | 31 Nedland Cres. | Pt. Noarlunga Sth. | S.A. Australia | 5167 | 08 386 1647 |

This table presents 7 fields and combines city and country in a single field. To create these, go to the table FILES (if the table isn't on the screen, type /) and type <enter> then OPEN to open a file. Anytime you make something with DM you open a file. Give this file the name DIRECTORY. Then go to the option EDIT with the arrow key and hit <enter> at ADD FIELD. You now have a screen which allows you to create the 7 indexes for the 7 fields. Here is the 7 fields which you create:

Record Size: 128

| Field | Size | Type | Description |
|---|---|---|---|
| 01 | 14 | 01 | First name |
| 02 | 16 | 01 | Surname |
| 03 | 30 | 01 | Address |
| 04 | 20 | 01 | Town |
| 05 | 20 | 01 | State/country |
| 06 | 10 | 01 | Postcode |
| 07 | 16 | 01 | Telephone |
| | 126 | | |

At the bottom of the screen you have the choice of the type of file to create. Choices: - Alphanumeric, which permits you to write letters and numbers. Hit (1) at the top of the page and then 14 for the number of characters which allow for 1st name. Under description you write "1st Name". To validate it, type <enter> and a second <enter> to end the 1st field. Do the same for the 7 fields...... if you make an error in typing then return with the up arrow. This also cancels the field in the option EDIT table.

DATA MASTER permits 18 records of 130 characters per line on screen. This is a distinct advantage on other systems. As the screen only covers 79 characters you need to use the arrow keys to read the complete lines.
How do we write 126 characters of our record to the screen?

The main points of location in our list are the names of the people on the list. You place them at the centre of the lines so that you always see the main data when you go from one end of a line to the the other, thus differing from the printout order above. I have chosen the format following.

SETTING UP THE SCREEN FORMAT

| Town | Address | Surname | First name | Telephone | State/country | Postcode |
|---|---|---|---|---|---|---|
| Pt. Noarlunga Sth. | 31 Nedland Cres. | Dalzell | Robert | 08 366 1647 | S.A. Australia | 5167 |

To make the above you need to go to the table TEMPLATES and type <enter> at EDIT DISPLAY. Here you have several choices of screen presentation format for your data. Type "1" and give this format the name of DIRECTORY. The screen changes and DIRECTORY appears at the top of the screen. Type <enter> to place the cursor on the screen. You then write "town" at the start of the line on the presentation screen. Instead of typing <enter>, type F1 and 04 to indicate that town is the field 04. You type <enter> and then you will see the space of 20 characters on screen for the name of the town. You then type address, F1 and 03. Then <enter>. You continue thus with: Family,F1.02 - 1st Name,F1,01 - Telephone,F1,07 - State, Country,F1,05 - Postcode,F1,06. Type C to return to a blank screen.

We can now verify if our format is the way we want it. To do this you have to at least write some data to the file. Now type /. Then go to the menu WORK. Type <enter> on ADD ITEM. Another <enter> will display the empty spaces on the screen to write our data to. Type:
PT. NOARLUNGA STH. - 31 NEDLAND CRES. - DALZELL - ROBBIE - (08) 386 1647 - SA.,AUSTRALIA - 5167.

To save the record you type YES. To "Another one?" type NO. Go to the menu OUTPUT and type <enter> on LIST ALL. Type <enter> again: you then see all the data on screen. With the right arrow you can move about the screen. Isn't it marvellous!

You have the power to change the data if you make an error, by placing the cursor at the error and then type F2 to make the corrections. The command F1 permits the lengthening or the shortening of entries.

If you now wish to take a few minutes break and leave the computer, then go to the menu FILES and type <enter> on CLOSE. Then go to the menu FILES and type <enter> on QUIT.

SENDING TO THE PRINTER
Return to the TEMPLATES menu and this time type <enter> on EDIT REPORT. After another <enter> you write to the first choice of format, the title DIRECTORY as before. You then make the different entries. Type for our DIRECTORY:

| | |
|---|---|
| Access Method | (1) |
| Line Width | (132) |
| Lines per Page | (66) |
| Margins: left | (4) - Top (2) - Bottom (2) |
| Number of lines of Replication | (1) |

Headings (3) - Detail (1) - Subtotals (0) - Totals (0)

Print CODES 27 15 for the NX-1000
27 20 for the DMP 106

On <enter> you are at the screen with the joint format complete. To put your DIRECTORY onto the screen again with the format printed in full. Begin with the heading TELEPHONE DIRECTORY which you type at TAB(55). On the second line you type at "1" Family name (ignoring the margin of 2).

| | | |
|---|---|---|
| 1 | Surname | (16 spaces - 16) |
| 17 | - First Name | (14 " - 30) |
| 31 | - Address | (30 " - 60) |
| 61 | - Town | (20 " - 80) |
| 81 | - State/country | (20 " - 100) |
| 99 | - Postcode | (10 " - 110) |
| 111 | - Telephone | (16 " - 126) |
| 127 | - File | ( 5 " - 131) |

You leave a line free then you type the printer codes of the fields next to insert "Field F1" as requested.

F1 02 enter F1 01 enter F1 03 enter F1 04 enter F1 05 enter F1 06 enter F1 07 enter F1 00 enter. Voila! Type F2 to save it. Then C.

We now come to the EDIT table view format to verify that our formats are good. If you want to print, then set the printer at 600 Baud and type PRINT FORMAT. To again see our printout format, go to the table OUTPUT and then type <enter> on VIEW ALL. To print the DIRECTORY to paper, switch on the printer and then type <enter> on REPORT. You Type "1" at the print option followed by 3 <enter>, and Voila! We have our report.

In addition to printing in the order of insertion in memory this programme will also print in the alphabetical order of your choice. For example, the three following:

| | | | |
|---|---|---|---|
| Access 1: | According to Surname | Field | 02, 01, 00 |
| " 2: | " " Town | " | 04, 02, 01 |
| " 3: | " " Postcode | " | 06, 02, 01 |

DATA MASTER permits you to choose the mode of alphabetical listing that you want as well as the one like DIRECTORY, which is according to the file number. To do this you go to the TEMPLATES EDIT ACCESS and you type "1" to write another; eg. according to the family name. You next specify the three selection choices of 5 characters each. Begin with 02 for the family name, 01 for the 1st name and 04 for the town. You then move to the last choice and enter 00. Next you go to the TEMPLATES EDIT REPORT and establish a new printing mode like you made for DIRECTORY. To print you go to OUTPUT REPORTS and, when you are certain that all the parameters are correct.

If you only wish to print the names commencing with the letters A and B then at OUTPUT REPORTS, SELECTION CRITERIA you type >=A AND <=B. This programme demonstrates the good value of the OS9 system and should encourage you to get familiarised with it. On the other hand, I am demonstrating that it is possible to use a programme in OS9 without knowing the system.
To do this it is enough for a Cocophile familiar with OS9, and having a knowledge of communication, to teach others the way to use such programmes.

DATAMASTER can be used to make lists of programmes, Magazine articles, books, records and a million other things.

Here is the final piece of information needed to finish off your session with DATAMASTER:
Return to the FILES menu and type CLOSE. To quit, type QUIT. To erase and destroy the file DIRECTORY, type DELETE.

END

COCO LINK

Graphics

# Kaleidoscope

## By Johanna Vagg

Here is a short Kaliadoscope programme to fill your television with interesting background graphics to complement the Xmas music you will be playing as you sit down to Xmas Dinner.

```
0 REM FOR COCO3
4 REM JO'S KALEIDOSCOPE
                    APRIL  1989
5 REM JOHANNA VAGG
    9 BELAH STREET
    FORBES NSW 2871
10 REM SPACEBAR TO CLEAR SCREEN
20 PALETTE0,0:HSCREEN2:HCLS
30 POKE65497,0
40 ON BRK GOTO 280
50 PALETTE10,32:PALETTE12,8:PALE
TTE8,34:PALETTE14,55:PALETTE15,3
3
60 X=RND(-TIMER)
70 CX=160:CY=96
80 C=RND(15)
90 X=RND(14)
100 MX=RND(140):MY=RND(70)
110 Y=RND(60)
120 HCOLOR C
130 HLINE(CX+X,CY+X)-(CX+MX,CY+M
Y),PSET
140 HLINE(CX-X,CY-X)-(CX-MX,CY-M
Y),PSET
150 HLINE(CX-X,CY+X)-(CX-MX,CY+M
Y),PSET
160 HLINE(CX+X,CY-X)-(CX+MX,CY-M
Y),PSET
170 HLINE(CX+X,CY+X)-(CX,CY+MY),
PSET
180 HLINE(CX-X,CY-X)-(CX,CY-MY),
PSET
190 HLINE(CX-X,CY+X)-(CX,CY+MY),
PSET
200 HLINE(CX+X,CY-X)-(CX,CY-MY),
PSET
210 I$=INKEY$:IF I$=" " THEN HCL
S:GOTO 80
220 X=X+4
230 IF X<70 THEN 130
240 IF Y>50 THEN 270
250 GOTO 80
260 CLS:GOTO 80
270 FORT=1TO30:FOR P=1 TO 15:PAL
ETTEP,RND(62)+1:NEXT:NEXT:HCLS:G
OTO 80
280 POKE65496,0:RGB:END
```

## ADVERTISING RATES:

$12.00 per full page
$8.00 per half page
$5.00 per Quarter page
MAX. 3 months

## Software:

Programmers Utility...............$20.00
(Reviewed Coco-Link No.2)

## Back Issues....$2.50each

CLASSIFIED ADVERTS

## WANTED

OVERHEAD PROJECTOR. Working or not.
Darren Ramsey (08) 384 6725

TANDY double sided disk drive for Coco 3 or trade single drive with cash adjustment.
Also wanted. SUPER VOICE song books volume 1 Potpouri and volume 2 Nursery rhymes.
Prices to:
Jim Eadsforth (08) 298 2843.

Multipak Contact: Robbie Dalzell (08) 386 1647

## COCO-LINK PD SOFTWARE

DISK 001 EDUCATION

1) Australian Geography
2) Australian Explorers
3) Fractutor
4) Decimal
5) Spellit
6) Times Table

DISK 002 EDUCATION #2

BINARY
COCOHOME
COINDEMO
FORMULA
MATCHEM
MATH
MATHSKT
MEMORY
NUMFUN
PUZZLE
TRIGSHOW
WORD

DISK 011 GAME

CoCo Trivia
Trivial Pursuit game.
(Takes up 2 sides of disk)

DISK 012 GAME

Computer Tote
Complete with races and tote betting. Marvelous for club fund raising!

DISK 013 13 GAMES

21 Card Trick
Bobo
Centrit
Germ
Max
Reversi
Yancc
25 Square
.Build
Cypher
Life
Maze
Tanks

DISK 014 11 GAMES

3Boxes
About
Memory
Patience
Puzzle
Wigworm
3Vagas
King Tut
Nausea
Pong
Slither

DISK 021 UTILITIES

3CLMLIST
3PRNTDOC
3QKMEN80
CATLOGUE
DSKDET
HASH
MULTUTIL
QKMEN32
3HBUFF
3QKMEN40
3VIPCOCO
DIRSORT
GOSUBBER
MENU
PRNTDOC

DISK 031 HOME APPLICATIONS

Homehelp
Budget
Will
Shoplist
Loan

ALL DISKS $5.00 each

Registered Publication No. SBH 1944

COCO-LINK MAGAZINE
31 NEDLAND CRES.,
PT. NOARLUNGA STH.,
S.A. 5167
(08) 386 1647