For your TANDY Color Computer

$3.25

AUSTRALIAN COCO

INCORPORATING MICO

December, 1984

Vol.1 No.4

For your TANDY Color Computer

incorporating MICO

January, 1985
Vol.1 No.5
$3.45

STRALIAN

C.C.

CFM COMPUTERWARE FOR MICROS.

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

NASHUA MD-ID 5¼" DISKETTES
40 Track S.S.D.D. Guaranteed 100% error free
Guaranteed 10 Million wear passed
TOP QUALITY AT SUPER PRICES
*10 diskettes in sturdy plastic library case $33.95
*Two pack $6.85
*Bulk with no write tab ..... $2.80 each

D x 85 PLASTIC DISKETTE FILE BOX WITH KEY
Holds 85 diskettes with dividers, coloured base with smoke lid ..... $28.00 each

DATALIFE head cleaning kit (kit cleans 2 disks) .. $10.15
DATALIFE replacement cleaning disks (10) $19.50

DATA CASSETTES
Contec CPF15 (c15) high quality data tapes.. $1.80 each
or box of ten $16.50
Clear case with metal spools

SUPER SPECIAL DISK DRIVES

CHINON F-051D 4OT
5¼" S.S.D.D. BARE DRIVE 250KB ½HT $184.95
CHINON F-502
5¼" D.S.D.D. BARE DRIVE 500KB ½HT $279.95
Power Supply and Case: Single Case $79.95
Power Supply and Case: Double Case $102.95
12 months guarantee against faulty workmanship
Technical Information Available...

PRO-COLOR-SERIES ©

FROM DERRINGER SOFTWARE INC.
Completely Updated Series
Arriving in January
Now available from C.F.M.
Pro Colour File $69.95
Pro Colour Forms $29.95
Pro Colour Dir $22.95
ALL THREE PROGRAMS $99.95

Requires 32K Disk Basic
SEE FULL ADVERTISMENT IN AMERICAN RAINBOW

ADD $4.00 HANDLING CHARGE WITH EACH ORDER
SEND $1.00 POSTAGE FOR CATALOGUE

SUPER BACK UP UTILITY©

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

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

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

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

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

*****ONLY $49.95!*****
(SUPPLIED ON DISK)

© M.L. BASIC COMPILER from WASATCHWARE — 64K REQ

* Convert your basic programs to M.L.

* You require no knowledge of assembly language. Although a competent M.L. programmer will save hours of work with this brilliant utility.

* Disk or Tape Compatible

* Compiles Colour Basic-Ext Colour Basic-Disk Basic-plus extra commands.

* Extremely user friendly program, backed up with an extensive 150 page user manual. Every function fully explained, including errors and what to do.

* No serious programmer can afford NOT to have this utility in his library.

Tape or Disk Versions $69.95

REQUIRES 64K EXT COCO

To: **COMPUTERWARE FOR MICROS.** P.O. BOX 81, BROOKLYN PARK, SOUTH AUSTRALIA 5032.

Mr/Mrs/Miss/Ms .....

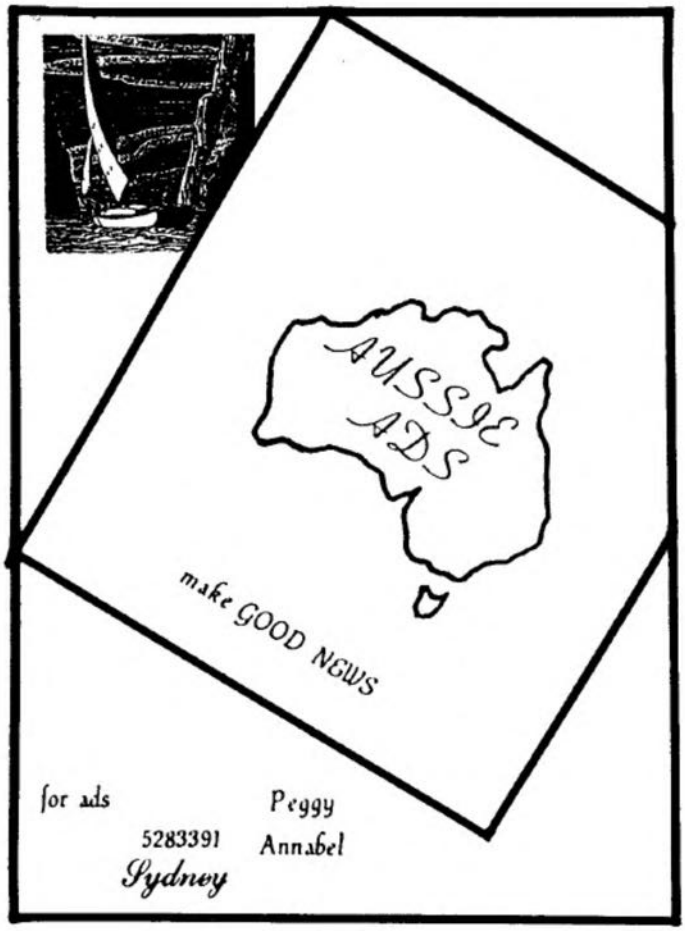
Address: .....

.....Phone:.....

VISA ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐

Bankcard welcome here ☐☐☐ ☐☐ ☐☐☐ ☐☐☐☐☐☐

PLEASE CHARGE MY ACCOUNT. Expiry Date ..... Signed .....



Now, for all versions of COCO, on disk or on cassette, the definitive EXPANDED COLOR BASIC and the BASSEMBLER

EXPANDED BASIC is designed to be added on to Extended Basic and includes the following new features:

- Printing text in all modes and colors
- A special 51x24 screen for PMODE4
- Scrolling of any screen section any way
- Extra colors in a new mode
- Borders for the text screen
- Extra graphics pages
- User definable sound effects
- REPEAT...UNTIL loops
- Multiline IF...THEN...ELSE statements
- Procedures in addition to subroutines
- Local and lower case variables
- Full ON-ERROR implementation
- Breakdisabling, auto-line numbering
- On-screen editing, 10 function keys
- User-definable characters /printer widths
- Single key entry of most BASIC words
- Ability to execute strings as commands.

The BASSEMBLER is a UNIQUE macro-assembler that allows you to inter-mix assembly language and BASIC in the same program - you can literally have one line assembly and the next line BASIC.
The BASSEMBLER understands the complete 6809 assembly language, macros and other pseudo-ops, labels and LINE NUMBERS too.
A disassembler/moniter Program is included with the BASSEMBLER.

Easy to follow instructions and demonstration programs come with the BASSEMBLER and EXPANDED BASIC.
64K is required. Specify disk or tape (giving model of recorder) and specify COCO model (1 or 2)

The BASSEMBLER	$15
EXPANDED COLOR BASIC	$30
EXPANDED BASIC + BASSEMBLER	$40
DISK USERS add an extra	$ 3

Buy now, direct from the author:
TINO DELBOURGO, c/- NIGEL MANN,
145 Nelson Road, Sandy Bay, Hobart, Tasmania.
7005. Phone (002) 25 1424

128K RAM CARD

Price not available at press time

INCREASE YOUR 64 K Co-Co OR Co-Co II TO 128 K RAM

- FITS COMPLETELY INSIDE YOUR COMPUTER
- SWITCHES TWO NEW 32 K BANKS OF RAM IN AND OUT OF MEMORY.
- BANKS CAN BE MAPPED IN THE UPPER HALF OR LOWER HALF, OR CAN ALSO BE A SECOND COMPLETE 64 K BANK.
- SWITCH TABLES INCLUDED.
- SIMPLE INSTALLATION AND DOCUMENTATION
- A MUST FOR OS-9 USERS.
- COMPLETE WITH 8 (4164) RAM CHIPS
- PAL CHIP HANDLES ALL BANKING COMMANDS.

OS9 SOFTWARE

Opack — Utilix — Filter Kit — SDisk & Bootfix — OS9 — Basic09 — Dynamite+ — Color Sleuth — RMS (Record Management System) — Stylograph Mailmerge — Spelling Checker Accs Receiveable and Payable Inventory (Control system for retailer and distributor)

AGENTS:
J Fricker Qld 07 262 8869
K Enright 09 36 28472
S Wilson Act 06 25 12339
Bayne & Trembath Vic & Tas
Ian Lee South Australia
L Wrigley 76 Molesworth St
Lower Hutt NZ

PHONE NOW

Blaxland Computer Services

(THE COLOR COMPUTER SOFTWARE SHOP)

P.O. Box 125 BLAXLAND 2774

047 39 3903

AUSTRALIAN COCO INCORPORATING MICO

Founder Greg Wilson

AUSTRALIAN EDITOR AND PUBLISHER
Graham Morphett

CO-EDITOR
Kevin Mischewski

EDITOR'S ASSISTANT
Sonya Young

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

COVER DESIGN
Jim Bentick

All Programs in this issue of Australian CoCo and MiCo are available on CoCoOz and MiCoOz

SEE CENTRE PAGE FOR DETAILS

DEADLINES

Feb 7th Jan '85
Mar 7th Feb '85
April 7th Mar '85
May 7th April '85
June 7th May '85
July 7th June '85

OS-9

Kevin Holmes is the contact for OS-9 information. He also has access to OS-9 Software from the U.S.
His address is:—
39 PEARSON ST., NARARA, N.S.W. 2250

Printed by:
Australian Rainbow Magazine
P.O. Box 1742
Southport Qld. 4215

Reg'd Publication QBG 4007

SCREEN DUMP

All of us at Rainbow hope that this Christmas will be a special one for you and your family.

We make this magazine because we care what happens in the world. We hope that, through the magazine, we can show that the love that is seen more often at Christmas can be a part of our lives right through the year.

Who needs all that weaponry ... the wars ... only the politicians and the generals!

Me, I'd rather sit on the beach with my wife and daughter, dig holes in the sand, and think of dumb things to write about in this magazine!

CoCo takes you wherever you want to be. If you need to be aggressive, you can shoot aliens from the other side of the black hole. If you need to obtain an insight into the mathematics of the solar system, CoCo will assist, if you want to teach your child anything from Bible stories to maths, music to programing (in several different languages), CoCo assists.

A glance at what we have in store for you in this magazine will show that in many ways, we are still only scratching the surface of CoCo's abilities.

In short, there isn't anything that can't be done in time with CoCo, yourself and preferably, a friend.

If you are not reading our magazine for the first time, you should know that this magazine is created by a group of friends, for a group of friends. It's a team effort, and we all take a great deal of personal pride in what we do here.

However, we traditionally get a whole new batch of readers at this time every year and to these folk we say welcome, hope you get a lot of pleasure from your computer and our magazines.

All of the articles are written by subscribers like yourself, who are kind enough to share their many talents with you. The articles are donated, and that makes them very special, particularly when you see the variety and quality each month.

INDEX

CoCoConf

GOLD COAST QUEENSLAND

JUNE '85.

PLAN TO BE THERE!

And speaking of friends, I know you will all want me to pass on to Johanna Vagg our collective congratulations on the birth of Katrina Lee, an 8 lb bundle of CoCo fodder for whom we've been waiting for what seems like years! Johanna is one of our regular authors, who despite the fact that she was about to have yet another child, still managed to get an article to us this month!

As our friends know, we've been stuffing up since mid year. Late magazines, late tapes and so on. BUT ... ta da ... we now have the Data Base under some form of control; as best as we can tell, we are now up to date with magazine mail out, and we have a big push on to complete all tapes before the end of 1984!

So let me introduce some of the team. I wasn't able to get the team from Brisbane down when I needed them, to get a more comprehensive photo, but then I also missed getting in the folk in Sydney, Melbourne, Hobart, Adelaide, Perth and many, many stops in between too. The photo shows a few of the more usual faces from around here, but as I have said, there are others ...

On the left, Jim and Sheryl; better friends it would be hard to find. Tireless workers, they have two little girls who didn't make it to the photo because they were asleep at the time!

The smoothy with the glasses, standing two feet above the rest of us is Kevin. Kevin, like Jim has a job elsewhere; when he finishes, he goes home to do a further eight hours for us! Kevin is ably assisted by his brother Glen, whose head appears just above Annette's (left) and Sonya's (right).

Annette is the finance wizz around here. She tells us what we can't buy. She is totally to blame for the fact that we are up to date. She works very hard and still manages to be an excellent mother.

Sonya is our most recent sucker friend. She has another job part time but will work with us full time next year. Sonya does good work.

Then there's me. That may look like a shadow from my hat, but it's really just one big black circle under my eyes.

Then there's Katie, the light of my life!

And finally, behind us all, is the Southport monument to better times.

Some interesting things are happening. The BIG news is that you can now obtain upgrades to 128K. As with 64K, this extra memory can not be used by CoCo's ROM Basic (or ECB).

However OS-9 can access the extra memory, as can (I think) Flex and perhaps one or two other systems. (You will recall from the article in last month's Australian CoCo, that OS8 provides it's own 128K upgrade.)

The sad news is that the latest CoCo has been having trouble with some TVs (no color). Rank Arena in particular seems to be a cause of concern. The problem is relatively easy to fix, and will not persist, I'm told....

Gathering strength are the voice packs - whether they come from Tandy, Software Spectrum, or the new REAL TALKER from Blaxland Computer Services, people are snapping them up! They are certainly a lot of fun, and will lead to a new approach to some Educational Software.

I believe that REAL TALKER puts a moving face on the screen as it talks to you - more next issue!

When we took over, one of the initial reasons for spliting the magazines was to be able to hold the price. The last thing I want to do is to raise the price - it hasn't risen in three and a half years. But its gonna have to happen. I looked at various "sweetheart" deals for old readers, the best that I can come up with is this:

If you are a current subscriber, or you can prove that you are a long term purchaser of the magazine, regardless of when you are due to subscribe, if you want to send along $28 per subscription for either RAINBOW or CoCo, or $56 for both, AND the money reaches here by 15th January, then we'll add 12 months to your current sub.

That's about the messiest way I've ever seen a business put its rates up, but then we're more a charity than a business anyhow!

There are obvious reasons this month why both magazines have the same material in the Screen Dump / Print#-2 columns, in fact it may happen every Christmas! Next issue the Magazines will revert to their more usual formats. RAINBOW seems to becoming a magazine for the specialist hobbiest, whilst CoCo reflects what you send!

Everyone who sent for Tee Shirts / Dresses has had their orders filled and posted. I'm sure that you'll like them!

"Faces at the Window" adds an additional dimension to CoCo this month. Fiction is something we get enough of from Tandy at times, but I thought that on this occasion, you might appreciate a little light reading over Christmas!

Finally, in case you didn't realise it, we've doubled the size of this issue. That's so we can have a holiday and just perhaps, not think about computers for a week or so. Annette, Katie and myself have planned a week in Sydney with our families, Sheryl, Jim and Co have a mad friend coming up from Sydney, Kev has to work, and Sonya is also off to Sydney. I'll have a good Christmas if those tapes are finished by then.....

LETTERS

Stephen Youngberry has a nice game in this month, and sent this letter with it:

Dear Graham,
Hope you like the game!

I tryed to put a play again routine in but it made the men twice the size and come apart. I wrote the game in PMODE 4, but changed to PMODE 2.

I also would like to find out if it is possible to change the cursor.

Stephen Youngberry.
Tara. Qld.

Dear Stephen,
The best way to 'run again' that I have found is to use a subroutine like this:

```
1000 I$=INKEY$:PRINT"RUN AGAIN?
Y/N";I$:IF I$=""THEN1000 ELSE IF
I$="N" THEN END ELSE IF I$="Y" T
HEN RUN
```

Check out OS8 this month for a way to fix (probably for good!) your cursor worries!

Graham.

Dear Graham,

Do you know of any article / book / publication, that covers in plain language, how to use the TRS 80 1.1 with an Epson RX-80 F/T printer for all styles and particularly graphics printouts, as well as with programmes such as Color Scriptsit, Screen Print, Color Logo, and Art Galley.

With Lotars Ginters last month, I would also prefer a combined CoCo magazine.

P. W. Coutts.
Kurrajong. NSW.

Dear P.....,

The program "GSPR", advertised for the last few months in both magazines will meet most of your requirements. The screen dump package mentioned (hopefully - Shery! hadn't written the article at the time that I wrote this!), this month will also meet your needs. Graphicon, reviewed in AUSTRALIAN RAINBOW last month, should also help.

We decided to run with two magazines, mainly because of the cost factor. When Greg decided to produce Australian CoCo, one of the pivital concepts was that it be separate, because he was mindful of the fact that a lot of our readers and contributors are young people. (Some, like Tino & Daniel Delbourgo, Tony Parfit, Andrew Simpson, Nicholas Birdsall and Rod Hoskinson are so good that you expect them to have spent more time with the computer than in fact they have!) Why separate the mags because of them? Because many of the kids coming along behind them, are unable to afford the price of a 'big' magazine, but are able to find the money to buy one of either RAINBOW or CoCo.
There was also that problem with the printer....

Graham.

Dear Graham,

I would like my name entered as a meet contact for

Dear Graham,

I was wondering if you could put me on the list of Contacts, to see if I can get some people interested in a monthly meeting. Initially it will be hard for me as I work shift work, but all things are surmountable, so "they" say.

My phone No is 057 98 1546.

I would like to wish you all a very, very MERRY Christmas and a great new year from myself, Elizabeth and daughter Alicia.

Peter Van Houts.
Violet Town. Vic.

Peter,

Thanks mate, hope you have some fun being the contact in Violet Town.

Grahan.

Dear Grahan,

The letter by Peter Miller from Tandy Maryborough, Qld. (October Aust CoCo) encouraged me to write. In the relatively short time that I have been associated with the Color Computer, firstly as a bumbling idiot, then as a bumbling novice, and now as a bumbling software agent for Software Spectrum, I have become more and more impressed with the CoCo.

Cudos go to these very dedicated people in my book. Tandy store managers in Victoria; Tony Chapman - Keysborough, Rob Berude - Airport West, (that name keeps on recurring - when is someone from Victoria going to introduce us properly to this man? G.), Eric Atkinson - Frankston and Bryan Mills - Forrest Hill. Tandy agents like Jack Smit at Sunbury and Bob at Bayne and Trembath in Rosebud are more than helpful.

Of course John Brothers and Kim Harper at Software Spectrum must be tops for support. One bloke was loaned a complete set of chips by John to fix a problem, all for the cost of VIP Data Base!

Now I'm all set. I subscribe to Australian CoCo & Rainbow, I buy American Rainbow too. I've ordered "Byte", "Help" and "Facts". I'm learning about "Flex" with my Tandy/J Dos Manual under my pillow. My EDITASM ROM is ready to go and I've worked out how to drive Telewriter 64 on my Gemini 10X at 9600 Baud. OS-9 is coming with Stylo and Dynacalc.

Just one question remains .... Will OS8 support any Cat's, Dir's, Ism's, Asm's, Mon's or Boots because I need all the help I can get before they take me away!

Doug Wilson.
Blackburn. Vic.

PS Does Jim Bentick I/O when my home town of Wonthaggi is mentioned as I went to school with one!

Dear Doug,

To complete your education, try Martha's CoCo cake, the recipe for which you'll find in this month's Rainbow.

Jim can be reduced to a simpering idiot very simply, just by uttering the word "Kernode".

Grahan.

Dear Graham,

I would like my name entered as a meet contact for the Pt Lincoln area. I have a 16K ECB CoCo 2 and I'm not an expert, but would help anyone that I could.

William Broadman.
Port Lincoln. SA. 086 82 2385

Dear William,

As I mentioned last month, you certainly don't need to be an expert to be a contact. Conversely, as a meet contact, you do have excellent backup any time you need it.

Well before I became the Editor of this Magazine, as a meet contact I'd need assistance with a problem and I'd ring around the contacts on the back of Rainbow. Never once did I ever get the impression that I was imposing - these people are great, and they will happily do anything to assist you - so welcome and I hope you get a lot of fun from being a Meet Contact in Pt Lincoln.

I didn't print the other part of your letter about Tandy. It's not that I don't agree entirely with you, it's just that right now they need all the help they can get!

Best wishes, Graham.

Dear Graham,

I am starting a User Group in Proserpine, it will be called "The Whitsunday Users Group". It is for Tandy computers as well as others. We will meet at the Proserpine State High School in our lunch hour as most of the group are students and a few teachers.

Could you please tell me if the group needs a permit. What discounts are the group entitled to, ie at Software Spectrum, Tandy, Color Burst Software.

Elden Foot.
Proserpine. Qld.

Dear Eldon,

To my knowledge, you do not require a permit to meet to discuss computers in Queensland yet, unless you want to have the group carry everything next door, then that could be misconstrued as a street march!

As for discounts, these have to be negotiated by yourselves with the relevant people. As a rule, it is unusual to obtain a discount from Tandy, other than for quantity buys, (we don't get one - let us know if you do!) and Software Spectrum operate in a simular manner.

Graham.

Dear Grahan,

I would like to make readers aware of some errors that crept into my program "WIZ LIST" (CoCoOz #17). My thanks to Roy Wynne of Melbourne for telling me about them.

1. It appears that the DELETE function does not work on the later DOS ROM models. A more rigorous check is made of the OPEN"O" statement. WIZLIST aborts at line 12010 with an AO error. The following changes should work with all ROMs:

```
115 PRINTIX,"wiz-list v1.3d10/11/84";:' Change
version
```

Delete the following lines:

```
DEL 690-710
DEL 890-930
DEL 12010
```

Change the following lines to become:

```
860 IF ANS="N" THEN940: ' CHANGES 840 TO 940
870 IF ANS=CHR$(12)THEN ANS="N": CF=1: GOTO940:
'Changes 840 to 940
950 PUT#1,J: 'CHANGE 2 TO 1
970 NEXTI:NI=J-1: NS=-(NI)NS)#NS-(NI(=NS)#NI:
IFLOF(1)=0 THENI=1
980 GOSUB2040:CLOSE#1
```

2. The "town" line can easily overflow if you do not know that there is a limitation to its length. This can cause the postcode to be truncated or to vanish.

Suburb, Town, State, Postcode together should be 24 characters or less. Since the Town does not appear on the printed label it is recommended that it not be used. This should provide adequate size for other fields.

3. A Search by Town or by Postcode does not work. This is fixed most easily by changing the following line:

In line 1980 change PT$ for PC$ and PC$ for PT$.

4. A search by surname does not work unless you put a comma at the end of the name to be searched. This is fixed by the following changes:

In line 1920 change LEFT$(NAME$,N-1) to LEFT$(NAME$,N-2)
In line 1710 change LEFT$(P$,N=!) to LEFT$(P$,N-2)

5. A PRINT of multiple labels per line with a search mask produces some duplicate labels. This is corrected by changing the following statement in lines 1200 and 1250:

Change KL=-(KL+1)#(KL(NL) to
IF FF()0 THEN KL=-(KL+1)#(KL(NL)

6. A list can sometimes mistakenly print. This is corrected by changing the following statement in line 1210:

Change IF KL)0THEN KL=KL-1 to
IF KL)0 AND LF()1 THEN KL=KL-1

7. The EDIT of the Town line mistakenly swaps Town and Postcode. This is fixed by changing line 500 as follows:

500 P$=PS$+"@"+PT$+"@"+PA$+"@"+PC$+"@"

I hope these errors have not caused subscribers too much trouble.

Sincerely,
"OZ-WIZ".
Brisbane. Qld.

Dear OZ,

Glad to see I'm not the only one having troubles with my Data Base!

Grahan.

Dear Grahan,

Greetings from Tassy. Australian CoCo is getting better every month. Congrats, really doing well. Reviews and Bulletin Board articles were excellent.

Any reviews of Data Base software programs for CoCo and how they work would be of interest to everyone I should think. Haven't read anything about that type as yet.

Have just got into VIP Writer and finding new and exciting areas.

Best wishes,
A. Jowett
DYNNYRNE. Tas.

Adrian!
You didn't see MY Data Base in September?!! Shame on you!

Grahan.

Dear Grahan,

I would be grateful if anyone can help me with a problem. It concerns the program "Hi-Res Screen Print Utilities". My copy of this program used to run perfectly on my Color Computer when it was 16K. However now that I have expanded my CoCo to 64K, the program no longer works. The program loads OK, however it will not run. Has anyone else encountered this problem, and if so, is there a solution?

S. Corrigan
LAWSON. N.S.W.

Dear S.... There does seem to be a small bug in this program - We'll see what we can find out.

Grahan.

Dear Grahan,

I think both the Australian Rainbow and CoCo are extremely excellent magazines. Keep up the good work!

B. Chenoweth
MELBOURNE. VIC.

Dear Grahan,

I've written to tell you about a bad mistake (well ... a little bit bad) in the game published recently in CoCoOz, ORBQUEST. I was made aware of this error while playing the game for fun. It is in line 51 which is a rather big line, so rather than repeat the whole line, I'll just point to the area that needs a fix:

51 ..... LO$(18)

should read

51 ..... LO$(19)="a bottle full of gas ":
.....

The correction is fairly important to the game later on. In fact it's hard to solve the adventure at all without this correction.

Tony Parfitt.
Wagga Wagga. NSW.

Tony,

People seem to be doing well with out the correction, we've had a lot of people say that they really dig your style!

Thanks Tony,
Grahan.

Dear Grahan,

Just a short note to say that your magazine has four ardent readers. Myself and my three boys, John, 16; Allan, 15; and Desmond 10.

Please keep up the good work as this is just what we need.

It is a pleasant change to read ads for Software for the CoCo here in Australia and not to have to worry about whether a program purchased overseas will work on our Aussie machines or not.

J.... Rae.
Mt Isa. Qld.

Dear J....

The really good thing about buying gear from our advertisers is that it helps me stay alive!

Grahan.

Just a note for some of our younger readers, and perhaps some who are not so young.

When you write to us, could you please ensure that you write your name and address legibly somewhere on the letter.

We really appreciate those who go to the additional trouble of including their subscriber number on all correspondance. Under the current system, we save five minutes every time we don't have to look one up. That can mean two hours some days!

Also, we have mail from several nameless people here, and seeing that we are now virtually up to date, if you feel that we haven't attended either to a sub or to a letter, then you may be one of the ones we are waiting to hear from again!

OS8

64K RAM ENABLE MODULE

This month we are pleased to bring you the RAM enable module which is the first thing you need to run the OS8 communications package. If you don't have 64K , don't read further.

The module, when loaded and RUN, will put the computer in 64K mode with full reset protection and scroll protection. You also get a ready prompt free of charge. The module also sets the map type to OS8 type 1, which is the required type for comms.

OS8 is able to go online in at least fifty languages with automatic translation from its 200,000 word language dictionary. Unfortunately, the dictionary is a separate package, and is commercial, so won't be presented in this column. However, it may be purchased from Rainbow Magazine for $49.50 including postage.

Alan Macintosh is working hard to get the comm package finished for next month, there was a problem with that garage company in Wee Wong, so in the meantime, get the RAM enable module up and running and we will see you next month.

The Listing:

```
10 'OS8 64K RAM ENABLE MODULE
20 'BY PAUL REVERE - NOV 1984
70 'SCROLL PROTECT - POKE 41805,
X WHERE X IS THE NUMBER OF LINES
 TO PROTECT FROM THE TOP
110 CLS:PRINT"ENTERING MAP TYPE
1"
120 CLEAR999
130 DATA 26,80,190,128,0,183,255
,222,166,128
140 DATA 183,255,223,167,31,140,
224,0,37,241,57
150 FOR I=1 TO 21:READ A:A$=A$+C
HR$(A):NEXT I
160 P=VARPTR(A$)+1
170 POKE P,126
180 EXEC P
190 POKE&HD6CD,&H00
200 POKE&HD723,&H14
210 POKE&HABEE,82:POKE&HABEF,69:
POKE&HABF0,65:POKE&HABF1,68:POKE
&HABF2,89:POKE&HABF3,62:POKE&HAB
F4,13
220 POKE&HABF5,14:POKE&HABF6,14:
POKE&HABF7,14
230 CLEAR200,32767
240 'SET RESET FOR USE WITH 64K
250 'FIRST READ STATEMENT IS A D
UMMY - SECOND NINE DATA BLOCK IS
 THE ONE THAT DOES THE JOB
260 POKE&HA055,&H0B:POKE &H72,&H
03:POKE&H73,&HF8:DATA 12,B7,FF,D
F,7E,C0,D4:FORI=1TO21:READ A:NEX
T I:FOR I=0 TO 6:READ A$:POKE&H3
F8+I,VAL("&H"+A$):NEXT
420 IFPEEK(&HFF22)/2 = INT(PEEK(
&HFF22)/2)THEN440ELSE 425
425 CLS:PRINT"OS8 64K RAM ENABLE
 MODULE 1.2.00";
426 PRINT"COPYRIGHT 1984 (C) AUS
T. RAINBOW"
430 PRINT"PRINTER OFF LINE"
```

LETTER

Dear Graham,

Following the recent article on OS8 (October, 1984.), I have searched high and low but have been unable to acquire a copy of OS8, version 2.1 at the recognised dealers.

I particularly want to be able to utilise the external memory access system, as I am finding my four floppies are more stringy than floppy these days due to the high speed they run at under OS8.

Can you please advise where this obviously very capable system can be obtained, and the price.

I.M.

Greenwood Park. WA.

Dear I.M.

In the first place, if you had read the article fully, you would have noticed that this product is unavailable until 1985.

However, as it turns out, a preproduction sample copy was flown in by the boys at Blacksland Computer Systems, and in due course I'm sure young Brice there will be pleased to let you know when you can have a copy.

Jackie Jackie at Parries Radio may also be in a position to assist you. Typically, the price of the system is expected to be in excess of $4000.00 for X/Mode 12 alone!

Graham.

VAGG 4

Johanna Vagg

FOURDRAW

My kids like 'drawing' programs. They were very enthusiastic about QUADDRAW - part of '4 for the Show' (April '84 Rainbow). Greg wanted programs suitable for the MC-10. So ... for my kids, for MiCo, and for non-extended BASIC, I wrote FOURDRAW. Instead of drawing in high resolution with the joysticks (as in QUADDRAW), you use the keys to draw (and change colour) in low resolution.

```
5 FOR M=1 TO 3
10 CLS0
15 K=2
20 A=0:B=63:C=0:D=63
25 W=12:X=12:Y=17:Z=17
30 SET(A,W,K):SET(B,X,K):SET(C,Y
,K):SET(D,Z,K)
35 A=A+1:B=B-1:C=C+1:D=D-1
40 IF A>23 THEN 50
45 GOTO30
50 W=W-1:X=X-1:Y=Y+1:Z=Z+1
55 IF A>29 THEN65
60 GOTO30
65 A=30:B=33:C=30:D=33
70 IF W=0 THEN80
75 GOTO30
80 K=4
85 A=0:B=63:C=0:D=63
90 W=14:X=14:Y=15:Z=15
95 SET(A,W,K):SET(B,X,K):SET(C,Y
,K):SET(D,Z,K)
100 A=A+1:B=B-1:C=C+1:D=D-1
105 IF A>25 THEN 115
110 GOTO95
115 W=W-1:X=X-1:Y=Y+1:Z=Z+1
120 IF W=4 THEN 130
125 GOTO95
130 K=3
135 FOR X=29 TO 34
140 FOR Y=13 TO 16
145 SET(X,Y,K):NEXTY,X
147 FOR L=1 TO 100:NEXT
150 NEXTM
155 FOR L=1 TO 500:NEXT
160 CLS:PRINT@33,"now it's your
turn to draw"
165 PRINT:PRINT" TRY USING 'W, X
, A, AND D'"
170 PRINT:PRINT" THEN TRY  'Q, E
, Z, AND C'"
175 PRINT:PRINT" FOR AN INTEREST
ING EFFECT          TRY (1-8)"
180 PRINT:PRINT" THE SPACEBAR WI
LL RUB OUT THE   LAST 'SET' IN E
ACH QUARTER"
185 PRINT:PRINT" AND '0' WILL RU
B OUT EVERYTHING EXCEPT THE LAST
 'SET'"
186 FOR L=1 TO 2000:NEXT
187 PRINT:PRINT"       any key to
 start"
188 P$=INKEY$:IF P$=""THEN188
240 CLS0
245 K=2
250 A=31:B=32:C=31:D=32
255 W=15:X=15:Y=16:Z=16
260 SET(A,W,K):SET(B,X,K):SET(C,
Y,K):SET(D,Z,K)
300 I$=INKEY$:IF I$=""THEN 300
310 IF I$="W" THEN W=W-1:X=X-1:Y
=Y+1:Z=Z+1
315 IF I$="X" THEN W=W+1:X=X+1:Y
=Y-1:Z=Z-1
320 IF I$="A" THEN A=A-1:B=B+1:C
=C-1:D=D+1
325 IF I$="D" THEN A=A+1:B=B-1:C
=C+1:D=D-1
330 IF I$="Q" THEN A=A-1:W=W-1:B
=B+1:X=X-1:C=C-1:Y=Y+1:D=D+1:Z=Z
+1
335 IF I$="C" THEN A=A+1:W=W+1:B
=B-1:X=X+1:C=C+1:Y=Y-1:D=D-1:Z=Z
-1
340 IF I$="E" THEN A=A+1:W=W-1:B
=B-1:X=X-1:C=C+1:Y=Y+1:D=D-1:Z=Z
+1
342 IF I$="Z" THEN A=A-1:W=W+1:B
=B+1:X=X+1:C=C-1:Y=Y-1:D=D+1:Z=Z
-1
345 IF I$="1" THEN K=1
350 IF I$="2" THEN K=2
355 IF I$="3" THEN K=3
360 IF I$="4" THEN K=4
365 IF I$="5" THEN K=5
370 IF I$="6" THEN K=6
375 IF I$="7" THEN K=7
380 IF I$="8" THEN K=8
385 IF I$="0" THEN CLS0
387 IFI$=" " THEN RESET(A,W):RES
ET(B,X):RESET (C,Y):RESET(D,Z):G
OTO 300
390 IFA<0 THEN A=0:B=63:C=0:D=63
:SOUND200,1
392 IF A>63 THEN A=63:B=0:C=63:D
=0:SOUND200,1
395 IFW<0 THEN W=0:X=0:Y=31:Z=31
:SOUND200,1
397 IF W>31 THEN W=31:X=31:Y=0:Z
=0:SOUND200,1
400 GOTO260
```

REVIEWS

SOFTWARE

SUPER BACKUP UTILITY

The Super Backup Utility (SBU) is a disk utility that provides the user with a number of options through a menu driven format. These options include -

1. Disk to disk.
2. Tape to disk.
3. Disk to tape.
4. Tape to tape.

Each program can be loaded and exec'd or may be loaded from the main menu.

DISK TO DISK

After loading the disk backup program, the user is faced with a number of options eg. number of drives, input drive, output drive, number of tracks etc. All of which can be answered with one or two keystrokes. Once these parameters have been set, all that is left to do is place the correct disk in the correct drive and away you go. That is of course with a two drive system. If using a single drive system you are prompted as to which disk to insert at the proper time. You have the option of one or more copies. Also something to note, the program requires that the destination disk be either bulk erased or unformatted before use.

There is also an option to allow J.Dos users to run this program.

Disk to Disk gives the user the capability of copying disks regardless of most protection schemes.

DISK TO TAPE/TAPE TO DISK

I found this particularly useful. If you have ever agonised over transferring machine language tapes to disk, it may be just the utility you have been looking for.

The user is first prompted as to which disk drive will be in use. Select the transfer you require. Make sure the correct disk is in the drive and that your tape recorder is either in the play mode (for tape to disk) or the record and play mode (for disk to tape). The first file name appears on the screen with either Bas or Bin denoting a basic or machine language program.

You are then prompted 'COPY Y, A, OR N'.

Y - Copy the progam, you have the option of the number of copies you wish to make (1 - 9) and whether you wish to go on to the next file or terminate.

A - Copy all programs including and following the one which is on the screen (with a nice space ~~in~~ between each).

N - go on to the next file.

During the transfer of any machine language program the start, end and execute addresses are displayed on the screen.

Also included is a relocate function which automatically relocates the start and execute addresses of programs that interfere with the disk operating system.

TAPE TO TAPE

Prompts once again let you know when to insert your tapes. One or more copies are available. The type of file, bas, data or m/l, is displayed on screen and so are the start, end and execute addresses of m/l files.

The tape backup uses no Extended Basic routines so can be used on computers that do not have Extended Basic.

This utility will also copy some protected programs for your own use.

Tape to Tape is available as a separate package. (on tape)

SPEED-COMP

Speed-comp included with the package is a machine language program for checking the speed of disk drives attached to the color computer.

Super Backup Utility is overall easy and simple to use. The prompts are quite staightforward and lead the user along, which means of course less time referring to the manual.

The screen displays are very clear and easy to read.

There are information and error messages to help identify any problems.

Included, as mentioned above, is a main menu from which you can choose the particular utility you require but to which, unfortunately, you can not return

from within the program. I found this rather superfluous.
The documentation is adequate, although some confusion may arise from the use of the terms 'Input and Output' as well as 'Source and Destination' disk/tape. It would have been better to stick with one version or the other.
This utility comes from Computize Inc. U.S.A and is available through Computerware for Micros of South Australia for $49.95. In my opinion, good value for money.

C Compiler

by Kenneth Ward.

INTRODUCTION.

The "C" programming language is a high level general purpose programming language which features economy of expression, modern control flow and data structures, and a full set of operators. It is very versatle and makes the TRS 80 Color Computer (CoCo) into an efficient, fast and more business like computer.

"C" was originally designed for and implemented on the UNIX operating system on the DEC PDP-11 by Dennis Ritchie. The language is available for use on IBM System 370, Honeywell 6000 and Interdata 8/32 machines which provides for some commonality / portability between different systems which was not available before on the CoCo.

"C" permits the creation of well designed, fully documented, and modular computerised solutions that are impossible with standard BASIC as supplied in CoCo's ROM.

OVERALL COMMENT.

In conjunction with "OS9", "C" provides the ability to design proper business solutions for implementation on the TRS 80 Color Computer.

It is easily implemented on the CoCo and can be learnt quickly due mainly to its simplicity of structure and the excellent layout of the manual. "The C Programming Language" by Kernighan & Ritchie.

"C" is implemented with a rich set of standard functions and library routines that enable the programmer to do almost all the functions that would be required to produce a business solution. If it is found that functions are missing then the programmer can:
- create separate modules for use with the standard library.
- incorporate 6809 Assembler code within the "C" code.
- create Macros either Assembler or "C".

NB. Incorporation of 6809 Assembler can reduce the portability of "C" programs.

The implementation of FLOAT, DOUBLE and LONG is poor and can lead to excessive execution times. Avoid these attributes or only use if absolutely necessary.

The missing functions (ref Product Review) do not greatly detract from the product. The functions can be easily coded and incorporated into the second library.

DOCUMENTATION REVIEW.

Manuals required.

- Microware C Complier User's Guide. (Supplied)
- The C Programming Language Brian W. Kernighan & Dennis M. Ritchie (The User's Guide states that this manual is provided (page 1-2) but it is not and the book can cost up to A$31.)
- Microware Relocating Macro Assembler Manual. (No details known).

The User's Guide.

Although called a 'user guide' this manual is very unfriendly. What it contains in techicnal detail, it more than lacks in legibility and structure. e.g.
- No installation information is given.
- Not one complete programming example of "C" for use on the CoCo.
- Few examples of how to use the Standard Library routines.
- Refers to nonexistant OS9 System Programmer's Manual.
- Refers to Microware Interactive

Assembler which is not supplied with OS9.

This manual contains a wealth of information and would be considerably improved if its format was changed to better present the information and provide examples as in the BASIC09 Reference and the OS9 Program Development manuals.

Differences from the implemation of "C" on the Co-Co and as documented in "K & R" are highlighted on half page, however the changes extend over most of chapter 1. Also "K & R" gives information concerning the implementation on the UNIX equipment and NO comparative difference is listed in this manual; e.g. the tutorial refers to the executable file "a.out" but one has to dig into the details of running the complier (page 1-11) to find that this file is not used; instead the current execution directory is used to store the compiled / linked program.

The assembler details in appendix D refer to the Microware Interactive assembler which is NOT supplied with OS9. Although the languages appear the same, it is most important that documentation references apply to the products supplied for OS9 and refer to the correct manuals.

C Programming Language

An excellent manual which was thoughtfully prepared and laid out. It explains the aims of "C" and abounds in examples and suggested programming exercises.

Also the manual suggests a methodology for good programming techniques and control flow.

PRODUCT REVIEW

Installation

Two diskettes are provided with the package:

1) "C" Compiler with 109 free sectors
2) "C" Library with 199 free sectors

The Compiler diskette only has a CMDS (commands) directory containing the "C" compiler along with the pressumably standard commands; copy, del, dir, echo and list.

To these commands I found it necessary to add; edit, rename, free, mdir, load and unlink, in order to reduce diskette movement. There is not enough space to comfortably hold the standard errmsg file for use with PRINTERR.

The Library diskette must be on drive 1 otherwise the linker will abort (ref. page 1-10). I, therefore created my data discovery on this diskette and set the data directory (CHD) to point to it.

Given the above setup it was now simple to edit files, compile, link and excute programs and when testing was complete I copied the finished promram(s) and source(s) to other diskettes for safe keeping and execution.

The examples for BASIC09 interface use the build command. As my typing is not that good I found it preferable to load and use EDIT.

I cannot stress enough the importance of BACKUP. The diskettes as bought must be copied and the copies used for your work. It is then just as important to backup/copy the work done.

Execution/Usage.

Well, what more can be said than in the Overall Comment section.

Once set up even my twelve year old son was programming from "K & R".

There are, however, severe limitations in using strings and mathematical functions. No mathematical functions are supplied and string manipulation is poorly supported.

The following functions are missing which are normally given with a basic package:

MID$, LEFT$, RIGHT$

ATN, COS, EXP, INT, LOG, RND, SGN, SIN, SQR, TAN

Also there is no support for graphics at all.

COMPARISON OF "C" WITH BASIC09

BASIC09

Advantages.

- Easy to code, has some similarity to ROM Basic
- Automatic differentiation of system keywords within the program code.
- Automatic indentation of program code e.g. LOOP and IF statements.
- Good standard set of functions.

Drawbacks

- Requires approx 23K to execute. (RUNB requires approx 12K). Thus with graphics (e.g. O-PAK) only 7K is available to develop the programs and 19K to execute them with RUNB.
- Requires line numbers to enable the ON ERROR function to be used. The line number cannot be changed with the BASIC09 EDIT command. It is necessary to exit BASIC09 and use the OS9 EDIT command to alter the line number without deleting and re-entering the whole line again.

"C" COMPILER

Advantages

- Good structured programs are possible and code (e.g. file structures) is only written once. This is due to the #INCLUDE facility.
- Easier data flow. Not only can parameters be used but also GLOBAL variables and pointers are readily available to use.
- Macros can be built to provide fast inline code for common functions. Thus with the aid of the OS9 Commands and OS9 Technical Information manuals it is fairly simple to provide the graphics functions that are not provided.
- Each program can be developed to use the 42K available under OS9 (if graphics are used - 35K).

Drawbacks

- Implementation of floating point arithmetic is abyssmal. It is slower than ROM BASIC and severely impeeds the performance of a program.
- Need to code some functions normally standard with a high level BASIC.
- Time to compile each program before testing can commence.

CONCLUSIONS.

It can be seen on close inspection of the two compilers that BASIC09 is aimed at
the "amateur" and "C" is aimed at the professional.

Once a programmer is trained in writing good structured code and proper use of documenting comments the advantages of an online debug package are reduced. Nonetheless, the debug serves to provide a better understanding of the functions of an application program and if the application is developed in accordance with current thinking using TOP DOWN design then each program fits properly into the application before any coding takes place.

Thus, with the exception of floating point arithmetic, the drawbacks of "C" are only minor and gladly suffered for improved performance of the final product. Only use FLOAT, DOUBLE and LONG if there is no other alternative.

The "C" User's Guide gives a very good summary on the BASIC09/"C" interface. The CoCo owner can therefore, if he has the money and time, use both compilers to their best advantage; e.g. BASIC09 for use with simple games, education packages, simple calculation programs as given in the American RAINBOW magazine for March 1984.

"C" would be used for the commercial programming of applications in place of Assembler as currently occurs for the normal ROM packages for the CoCo.

Comparing Other "C" Compilers.

A series of articles appeared in the BYTE magazine in August 1983 concerning "C" compilers and their use on 16 bit machines (8086 and 68000).

I was able to use directly the benchmarks for floating point, Eratosthene's Sieve and a Fibonacci number generator for "C" on the 6809 and it was easy to adapt "float" and "sieve" to use with BASIC09 and ROM BASIC.

COMPARISON of EXECUTION TIMES

All programs loaded into memory before execution.

	ROM BASIC	BASIC09	"C"
Example 1	122	30	7 (1.7 if pointers used)
Example 2	317	86	15
Example 3 Floating point arithmetic (from BYTE August 1983)	98	45	155
Example 4 Eratosthene's Sieve (from BYTE August 1983)	31	31	3

All times in seconds.

Example1

```
100B=0
110FOR I=1 TO 10000
120A$="ASSIGN STRING"
130B=B+1
140GOSUB 200
150NEXT I
160END
200RETURN
```

Example2

```
100B=0
110C=1
120FOR I=1TO10000
130A$="ASSIGN STRING"
140B=B+1
150B(C)=B(C)+B
160A$(C)="ARRAY"
170B$(C)=A$(C)+A$
180GOSUB 250
190NEXT I
200END
250RETURN
```

The results for comparing "C" with BASIC09 and ROM appear on the preceeding page. Below is a table setting out my results on the 6809 with a comparison of the figures taken from the article "Comparing C Compilers for CP/M-86" in the August issue of BYTE.

Thus it can be readily seen that for Char and INT work the combination of the 6809 with OS9 and "C" gives a result that compares most favourably with 16 bit processors running at a clock speed of 5MHz or above. The IBM PC runs at 4.77MHz. Also even though FLOAT is slow, its overall performance to the 8086 CPU's is reasonably good. This also suggests that the BASIC09 arithmetic is very good.

The price of "C" is $149.95 and it is available from all(?) Tandy stores.

Program	CP/M times maximum secs	CP/M times mimimum secs	6809 time secs	Ratio CPM / 6809 maximum	Ratio CPM / 6809 minimum	Effective Program Size maximum CP/M	Effective Program Size minimum CP/M	Effective Program Size 6809
Sieve	12	3.6	24	0.5	0.15	416	128	210
Fibon	25	14	111	0.23	0.13	176	112	154
Float	278	95	1550	0.18	0.06	No figures to compare		

(Effective program size is the size of the program with all its instructions minus the size of a C program with only the bare essentials.)

GAMES

Return of the JET-I

One of the many highlights of the movie Return of the Jedi was the scene were Luke and the princess zapped through the forest on their rockets persued by the enemy.

It was great right? Well, Return of the JET-I brings it all back, but this time, you're in the driver's seat (not sure 'bout the princess tho').

After loading, press the fire button on either joystick and select the number of players.

A title page displays each players score and highest score of game.

There is a skill level scale, which each player can choose from. My four year old had no trouble with levels 1-3, so I suggest if you want a bit of excitement look up tuther end of scale.

Begin the game seated on a spacecycle, you have six shields that can absorb six hits, a bar graph above indicates shield strength, on the seventh hit it's kapoot, so don't be generous with those shields.

As you zip through the forest you must try your darndest to avoid the trees that came hurtling at you. When the enemy appears you have two ways in which to dispose of them and score points. Either nudge them into the nearest tree, that's always nice, or, you can shoot them with your laser as they try to escape (I like that one). Extra points are gained if the enemy rider is a long way off.

Points are also awarded by snatching up the little bears (remember them?). Each bear is worth 1000 bonus points.

If you make it through all the mayhem you now have the emperial power station to be penetrated and then knock off the power beam.

Through the walls (this I had trouble with, I tend to get the wobbles just as I'm about to pass through) then onto the radar dish set right on top of the main building, a direct hit will destroy the power station and add one segment to your shield strength.There you did it.

To summaries, some good points, great range of skill which gives everyone a go, from the littlies to whoever (with the higher levels you really stretch) JET-I plays well, the graphics are fair, but the end, what a let down, very whimpy.

A postscript. If it is possible try and shop personally, The blurb provided states the distributors take no responsibility for any damage during shipping, with the standard of postal service around. . .

RETURN of the JET-I on cassette from Thundervision and is distributed nationwide. The price, $30.00 but for December only it is available directly from Thundervision for $25.00.

32K GAME.

Tandy have just released "Pooyan" on tape. Pooyan may well rank as the best game for CoCo yet.

The graphics are superb, the colors are better than usual, the music is at least as good as anything you've heard CoCo produce, and it plays very well.

This is one game that plays better with the keyboard than with joysticks, however you can use either joysticks or arrow keys and spacebar to defend your otherwise defenseless piglets from the wolves who float down from a near-by tree to attack your home and family.

There is stratergy to be employed, and skill to be acquired. As you get better, the skill levels change automatically.

At $49.95, the game is not cheap, but quality always bears a premimum. A few more of these Mr Tandy and you'll be cookin' with gas!

Workspace

If presentation can be used to measure the quality of a software package then Platinum Software's WORKSAVER has a head start with its well presented package of three printed manuals and keyboard overlay. But as one would naturally hope the software that makes the package certainly measures up to first impressions.

What is WORKSAVER? It can be best and accurately described as a Programmers Utility Package. It provides a BASIC programmer with a multitude of utilities to simplify and speed work.

Microsoft BASIC as used on the Color Computer is a very powerful and flexible programming language, but when TANDY designed the Color Computer they chose to keep the language as simple to use and compact as possible.

WORKSAVER among other things adds two very powerful features. Firstly we have a Dynamic Screen Editor in preference to our normal line orientated Editor. Secondly we have single key entry of all the BASIC commands, a system MC10 users know well.

Why should you want to use a screen editor in preference to our normal BASIC editor. A screen editor allows you to move the cursor anywhere on the screen and to change the character under the cursor simply by retyping over that character. There are many additional powerful commands, including erase to end of line, erase to start of line, disable overtyping and insert, move cursor to specified character and more. All these commands are executed using simple two control key operations.

This control and function key action is extended to the entire keyboard to allow rapid entry of any BASIC command. The keyboard overlay supplied with the package fits neatly over the old grey case type keyboard and has printed on it the commands relating to each key. Unfortunately the overlay will not fit over the new CoCo2 keyboard so you will probably have to settle for using it purely as a reference table. Both the BREAK and CLEAR keys are used as control keys, so in effect you are redefining the entire keyboard 3 different ways. That is before you even consider that WORKSAVER allows you to define over 100 of your own programable key arrangements.

LIST ON ERROR causes your program to list to the screen with the cursor at the error location any time that an error is encountered while running. BASIC will still print out its normal error message so that you know what error you are looking for.

A number of other simple yet 'necessary' features included in the WORKSAVER are the ability to selectively scroll portions of the screen, AUTO LINE NUMBERING, relocating single lines or groups of lines, the ability to echo the screen output to the printer and further, global search and replace. This last feature is superb. Perhaps you find that instead of typing in LINEINPUT you used only INPUT through your entire program and wanted to change it. With WORKSAVER all that is needed is to ask for it to find every occurence of INPUT and replace it with LINEINPUT, easy huh! But that's not all, what if you want to leave some of the INPUT's as they are? You have the option of selectively searching and replacing the chosen string. Of course you may exit the search at any time simply by pressing BREAK.

Disk users are well looked after. DIRPLUS is an additional program included in the package providing several disk related utilities. Firstly is an enhanced directory listing, which inludes in the listing a complete Granule Allocation with each directory entry. Anyone who has attempted to rebuild the Granule Allocation Table from a crashed disk will know how valuable a printed copy of the granule alloction for each file is. DIRLIST also prints out the START, END and EXEC addresses of any machine language programs.

Of course when you are ready to repair that crashed disk WORKSAVER is ready with DISK EXAMINE which allows you to examine any sector on the disk, a byte at a time using WORKSAVER'S screen editor.

Worksaver is undoubtedly an excellent package, well presented. Everything I tried worked as the manual explained with very little difficulty. Any problems I encountered were quickly solved through reading more carefully the appropriate sections of the manual. To do full justice to the program would require that you read the manual thoroughly and experiment and experiment with the commands, but if you are at all serious about your programming then I have no doubt that WORKSAVER will make your work so much easier.

I understand that WORKSAVER is fully compatible with all versions of the Color Computer both 1.0 and 1.1 and is available to suit all systems from 16K non-EXTENDED BASIC to 64K DISK BASIC.

WORKSAVER is availiable from:

GEOFF TOLPUTT
PLATINUM SOFTWARE
P.O. Box 140
Wooloongabba, QLD 4102.

Price:

Tape WORKSAVER PLUS	$50.00
Disk WORKSAVER PLUS	$60.00
DISKPLUS	$15.00
P&P	$3.00

Printer Interface.

The Fiala Connection.

Bob Thompson reckons that he has finally found something he likes for CoCo .... no ifs or buts.

I found that hard to believe - I mean he's a nice enough guy and all that, but nothing is ever good enough!

Then he told me about the Fiala printer interface and I understood. Geoff Fiala has been selling his interface for several months now, long enough for word to be filtering back that this product is good.

The product is very professionally made, has none of the "hackers" look about it, and will work between 300 and 9600 baud. At $75.00 it's too cheap!

Available from Geoff or Blaxland Computer Services.

Diskettes.

Greg gave up on Diskettes. He sold them for as long as he dared, then stopped. Even for his own use the diskettes were not proving satisfactory.

But the enquiries keep coming, even to this day, and as we need a supply here in any case, we looked around for something that would work for us as well as those who wanted to buy them here.

We wanted to be able access both sides, whether the disk was refered to as "double sided" or not.

Any disk we took on had to represent good value for money.

We wanted to be able to use the disks for 80 track work as well as 40.

We decided on DATA PARTS Diskettes and so far have been very pleased with them.

The quality appears to be there, they handle all that we throw at them, and they are only $28.95 a box, or if you want to be cautious, $3.65 ea.

Apart from ourselves, you can buy these diskettes from Blaxland Computer Services, PO Box 125, Blaxland. NSW, 2774, or Data Parts, 11 Edward St., Shepparton, Vic, 3630.

FANTASTIC FLEX

with William .Flexgerald

Part I

There are three main operating systems available for the Color Computer. They are TRSDOS, OS9, and FLEX.

This column, which will appear on a semi regular basis, is dedicated to the FLEX Operating System. I use all three operating systems, and it is not my intention to denigrate the other two, but to pass on some acquired knowledge of Flex.

To start with, Flex is a very user friendly and memory efficient operating system and has been around for a long time. This means that the software base for this system is immense. Apart from the applications software available, the public domain has contributed an enormous number of utilities. Most of these have been dealt with before and I won't cover old ground. This column will try to explain some of the more interesting applications available under Flex. By the way, most of these applications are or will be available under OS9.

The first program I will discuss is Record Management System, or RMS for short. RMS is a true database manager with its own programming language, and can be successfully used for moderate business applications.

The purpose of a database is to store and retrieve information as efficiently as possible. RMS does this in a way that may frighten the newcomer, but will please the seasoned user immensely. A file manager like PFS will give you prompts all the way through the setup procedure, but it is this very friendly structure that makes the file manager so inflexible. Anything you need to do outside of its expected pattern is not possible.

A database manager on the other hand, does not prompt you to nearly the same extent when you are setting it up, but will allow you to format it any way that you can think of. This initial formatting of the database is an acquired skill and needs a reasonable amount of database programming knowledge. Once done though, the finished database can be so simple to use that an untrained person can master in in less than an hour.

The true strength of RMS lies in the record format which creates the screen fill out form. This record format can be specified to validate every piece of data entered. If a staff member tried to enter alphanumeric information in a money field or date field, they would simply be told to do it again, and the input would not be accepted until it conformed with the required field validator. This makes RMS a very safe database in the hands of untrained clerical personnel, as it would be very hard for them to accidently destroy any previously entered data. In fact, it would be pretty hard to enter ANY bad data at all.

The second strength of RMS is the Report Writer. This very flexible program is responsible for most of the output to printer, screen or disk, and can be programmed to output data in an infinite variety of formats. Apart from formatting, Report controls the inclusion or exclusion of specified records or groups of records. This information is written into each individual report file, and you

can create any number of report files for specific data retrieval requirements.

The same is true of INDEX files. An Index file is simply a sorted text file containing the first word of each record's selected sort field. Once again, there may be as many of these different index files as you require, and you may mix and match them with the report files as you wish.

Rather than do a review of RMS, which has been done before, I will present a sample database layout for you to type in and modify for your own needs. This database was a skeleton written for a friend in the medical profession, but can be changed with a text editor for home or business use by simply changing or adding fields.

The first file that needs to be created is the dictionary file. This file contains your records format, validators, and on screen prompts, and can be created with a word processor or the FLEX text editor. If you don't have these, the BUILD utility can be used. It is best to use a word processor like STYLOGRAPH as the other two do not allow you to use lower case in the prompts. Here is the the dictionary file:-

```
"                    MEDICAL DATABASE -
ACCIDENTS";
NAME 20 A "Patient Name: ";
ADDRESS 40 A "Address: ";
AGE 3 N* "Age: ";
DATE 8 D "Date of injury: ";
FINGER 20 A "Injury: ";
AMPUT 2 A "Amputated (Y/N)? "
[Y,N,];
CAUSE 30 A "Cause: ";
$
"                    PATIENT    SUB-
RECORD";
NAME 20 A "Name: ";
TDATE 8 D "Date of treatment: ";
DOC 20 A "Treating Physician: ";
FEE 8 M "Fee Charged: ";
MED 10 A "Medical Plan: ";
COMM 57 A "Comments: ";
$
```

Let's take a look at the dictionary file line by line. The first line is the database name and must be present and contained on a single line. In this 34 column format it appears on two lines, but I stress, it MUST be typed in as one line.

The second line must always be a record format line and designates the key field in the primary record - NAME is the field name, 20 is the field length, A designates alphanumeric information, and "Patient Name"; is the on screen prompt. The following six line follow the same general format. The N* after AGE designates a numeric field with the * sign allowing you to leave the field blank. This is normally not permissable as a safety measure.

The D in the DATE field validates that the information entered is a proper date in American format, and this type of field must always be 8 characters long. The [Y,N,] in the AMPUT field is a validator and will allow input of only Y or N. Each line must end with a semi-colon or an error will occur.

The $ sign after the CAUSE field indicates the start of the secondary record format and must be followed with a heading. The first secondary record field should always be exactly the same as the first primary field as this is a common key field and copies the information in the primary key field. The other field names should be different from the primary names, but the prompts can be the the same if nescessary. (Not recommended). The final $ sign signifies the end of the secondary record. When this is complete, save the file to disk with your own filename - this must have an extension of .DIC

Total record length can be a maximum of 250 chars. for each primary and each secondary record, and seeing as how you can have many secondary records attached to each primary this can be quite wasteful of disk space. The next

the Pat Kermode

Like Johanna Vagg, I am another CoCoNut mother. I love reading Johanna's articles, and her efforts prompted me to write this article.

Firstly however, I am hoping someone can help me!

In "HEIST" (AUST RAINBOW July 1984, No37, Page30), how do I peek the locations, to find the FC ERROR, as Johanna said she did with "CoCo COMPOSER"?

Enough of my problems, on with the show!

My nine year old daughter loves to type in short programs. Recently, after a Sunday morning sleep in, we woke to find her playing with a "DRAW" program we knew we didn't have on tape. (We knew this because we had been waiting for a certain someone in Queensland to send it, and waiting, and waiting .....) Anyway she had typed this program in while we slept.

I looked in computer shops for more of this type of program, but any available suffer from too much typing in for too little reward, or they tend to be trying to teach BASIC programing and tend to be boring.

I therefore decided to get together a collection of short fun programs for children to type in themselves.

Here are a selection of these programs:

1. THE AMBULANCE SIREN.

```
10 FORX=1TO50
20 SOUND175,5:SOUND100,5
30 NEXTX
```

(There are all sorts of variations on this theme, I have found the higher notes more effective, as in:

```
100 SOUND220,4:SOUND195,4
110 GOTO100
```

Use these two programs to experiment, you can achieve some very good results with very little trouble. G.)

2. TABLES.

It really isn't fair! Five years ago, when I was at school, I could have used this:

```
1 '******TABLES****** BY the PAT
 KERMODE******11/84
10 CLS0
20 INPUT"WHICH TABLE DO YOU WANT
 TO TRY";Z
30 FORX=1TO12
40 PRINTTAB(10)Z"X";X;"=";Z*X
50 NEXTX
```

(I interfered with Pat's design, and modified it to produce a slightly different result, just to show how, with relatively few commands, you can get CoCo to do some quite clever things.

This is my version of Pat's program:

```
1 '******TABLES****** BY the PAT
 KERMODE******11/84
10 R=RND(9):R=R-1:CLS(R)
20 INPUT"WHICH TABLE DO YOU WANT
";Z
25 IFZ=0THENRUN
30 FORX=1TO12
40 PRINTTAB(10)Z"X ";:PRINTUSING
"##";X;:PRINT" =";:PRINTUSING"##
#";Z*X:SOUNDX*Z,2
50 NEXTX
60 PRINT@451,;:LINEINPUT"PRESS <
ENTER> TO CONTINUE";A$:RUN
```

Notice how I dont tell you what the differences are? Type the first program in, run it and then modify it to this one. The best way to learn, is to type, type and type! - G.)

3. FACE (ECB - Eyes changing with POKE178 patterns.)

```
1 '*****FACES***** BY the PAT KE
RMODE******11/84
5 PCLS
10 PCLS8
20 PMODE3,1:SCREEN1,1:COLOR5,8
30 LINE(120,90)-(140,110),PSET,B
F
40 LINE(40,140)-(220,160),PSET,B
F
50 X=RND(255)
60 POKE178,X
70 LINE(50,20)-(100,60),PSET,BF
80 LINE(160,20)-(210,60),PSET,BF
90 FORY=1TO500:NEXTY
100 GOTO50
```

4. CITY SKYLINE. (ECB - Is different each RUN.)

```
1 '*****CITY SKYLINE***** BY the
 PAT KERMODE ***** 11/84
10 PMODE3,1
20 PCLS
30 SCREEN1,1
40 FORS=1TO15
50 X=RND(255):Y=RND(191)
60 LINE(X,191)-(X+RND(75)+20,Y),
PSET,B
70 FORX=1TO200:NEXTX
80 NEXTS
90 I$=INKEY$:IFI$=""THEN90 ELSE
RUN
```

Notice how you can use INKEY$ to postpone an event. - G.

5. COUNTING. (by 2's, 3's, or whatever.)

```
1 '****** COUNTING ****** BY the
 PAT KERMODE ******11/84
10 CLS0
20 INPUT"INPUT NUMBER TO COUNT B
Y";S
23 CLS0:PRINT@8,"COUNTING BY";S;
"'S";
30 P=X*S
40 PRINT@238,"       ";:PRINT@238
,P;:SOUNDRND(255),5
45 A$=INKEY$:IFA$=""THEN50 ELSE
RUN
50 FORA=1TO200:NEXTA
60 X=X+1
65 PRINT@451,"PRESS ANY KEY TO R
ESTART";
70 GOTO30
```

6. JOYSTICK DRAWING. (Right joystick, BREAK to END.)

```
1 'JOYSTICK DRAWING***** by THE
Pat Kermode******11/84
10 CLS0
20 PRINT"DRAW SLOWLY WITH YOUR J
OYSTICK";
30 PRINT@64,"PRESS JOYSTICK BUTT
ON TO CHANGE COLOR";
40 PRINT@480,"PRESS ANY KEY TO S
TART";
50 A$=INKEY$:IFA$=""THEN50
60 CLS0
70 H=JOYSTK(0)
80 V=JOYSTK(1)
90 IFV>31THENV=V-32
100 IFH>63THENH=H-63
110 SET(H,V,A)
120 P=PEEK(65280)
130 IFP=126THENA=RND(8)
140 GOTO70
```

7. STORY.

This is a longer program and allows you to type in a story and then have it scroll across the screen for Mum and Dad, or whoever to read.

After typing "RUN", you are asked for your name, and then the name of the story. You are then prompted to write your story of up to 5 paragraphs, each of less than 7 lines long.

The use of LINE INPUT allows for use of commas, inverted commas, exclamation marks etc.

Scrolling can be slowed by changing the 150 in line 310 to 300 or more for very slow readers.

If Graham will permit, (I will! - G.) I would like to ask you to submit ways to save the story to tape. If we're extra nice to the old fella, he might print the best answers next month! (Wot's with the 'old'? - G.)

THE LISTING:

```
1 '****STORY**** by THE Pat Kerm
ode**** 11/84
10 CLEAR1000
20 DIMP$(7):P$(0)="##############
##":L$="      ":S$="            "
30 CLS0
40 INPUT"WHAT IS YOUR NAME";N$
50 CLS0:PRINT"WHAT IS THE NAME O
F YOUR STORY"
60 LINEINPUTT$
70 CLS0:PRINT"YOU WILL BE ASKED
TO TYPE UP TO FIVE PARAGRAPHS. T
HE PARAGRAPHS  MUST NOT BE LONGE
R THAN SEVEN   LINES; OR THEY WI
LL NOT COME UP ON YOUR SCREEN."
80 PRINT@224,"START TO TYPE IN Y
OUR OWN STORY.********FIRST PARA
GRAPH*********";
90 LINEINPUTA$
100 P$(1)=L$+A$+S$
110 PRINT"*******SECOND PARAGRAP
H*********";
120 LINEINPUTB$
130 P$(2)=S$+B$+S$
140 PRINT"********THIRD PARAGRAP
H*********";
150 LINEINPUTC$
160 P$(3)=S$+C$+S$
170 PRINT"********FOURTH PARAGRA
PH********";
180 LINEINPUTD$
```

Dear Doctor CoCo

Dear Doctor CoCo,

I have recently upgraded to Disk Extended Basic and have found that some basic programs from various sources, mainly CoCoOz, that worked before, seem to hang up the computer. Can you give me a clue as to why this might happen,

Susan Williams,
Byron Bay. N.S.W.

Dear Susan,

Many basic programmers have taken advantage of a poke to slow down screen output and change it to amber. This gives their programs better presentation and works fine on Extended Basic. On Disk Basic, this poke causes the computer to apparently hang up. To get out of the hangup, 1. Press Break 2. Type Poke 359,126:Screen 0,0. Nothing will appear on the screen as you type this, but it will normally return you to the standard display.

The poke that causes this problem is Poke 359,60. The best fix is to go through the program and change all occurrences of Poke 359,60 to Poke 359,57 and save the program as modified.

Dear Doctor CoCo,

When I try to hand assemble machine code using one of the short programs from Rainbow, sometimes it works and other times the program crashes What is doing this?

John Makepeace,
Darwin. N.T.

Dear John,

The Basic loader you are using is a common one and quite effective if you are typing in a machine program located above $3000.

On power up, you are in Pclear 4 mode and the basic loader will be automatically located at address $2601 and will occupy a few hundred bytes after that. If your machine code program occupies any of that space, it will overwrite your basic loader as our are poking it in and cause it to crash. Check the start address of the ML program in your listing and if it is going to conflict; do a PCLEAR 1 before loading the Basic hexloader. You could also do a PCLEAR 8 if your ML program is going to use low memory and that will put the hexloader past $3000 in memory..

Dear Doctor CoCo,

Can you please send me the nescessary circuit modifications to interface the new Tandy sculptured key board to the PIA on my old grey CoCo (E board).

Len Maloney,
Mackay. Qld.

Dear Len,

According to informed sources at Tandy, the kit supplied with the new keyboard is only required to interface with the older American models, which had a different PIA from any of the Australian Computers. These are the models up to the D board. If you have and E board, the keyboard will plug straight in, providing you have the appropriate socket on the keyboard connector. For an F board computer, the standard fish tail type socket that comes with the keyboard will plug in directly. The E board however, requires an older type socket connector with a row of brass pins. This connector must be mated to the flexible tail coming from the keyboard. John Brothers from Software Spectrum has been known to supply these connectors on the odd occasion, and if you speak to him nicely, may be able to assist you. The actual jot of mating this connector is quite difficult and requires expert handling, but I have seen it done successfully.

Kermode Kont.

```
190 P$(4)=S$+D$+S$
200 PRINT"********FIFTH PARAGRAP
H*********";
210 LINEINPUTE$
220 P$(5)=S$+E$+S$
230 P$(6)=S$+"*THE END*"+L$
240 W$="***THIS IS "+N$+"'S STOR
Y***"
250 CLS2
260 PRINT@128+(32-LEN(W$))/2,W$;
270 PRINT@192+(32-LEN(T$))/2,T$;
280 FORX=0TO6
290 FORA=1TOLEN(P$(X))-15
300 PRINT@328,MID$(P$(X),A,15);
310 FORD=1TO150:NEXTD
320 NEXTA,X
340 RESTORE
350 PRINT@384,"**DO YOU WANT THE
 STORY TO RUN  AGAIN Y/N";
360 INPUTR$:IFR$=""THEN360
370 IFR$<>"Y" AND R$<>"N" THEN36
0
380 IFR$="Y"THEN240
390 END
```

Basic Tutorial

I/O, I/O, AND OFF TO WORK WE GO.

Conversation with CoCo owners suggests that a significant percentage have problems with data and software storage using tape. Most of the complainants are new users, suggesting in turn that a little knowledge and experience go a long way towards eliminating these difficulties. This article is therefore directed towards the new user, and may be timely for those blessed by a new CoCo for Christmas.

by John Poxon.

THE CASSETTE SYSTEM.

When considering the cause(s) of input or output (I/O) errors, the first item on a list of possibles must be the cassette system and it's condition.

A cassette system, in its simplest form, consists of a suitable audio tape recorder, in good condition, connected to the CoCo by an appropriate cable assembly. "Suitable" means a recorder without sophisticated signal processing gimmickry. A plain, low cost unit such as a Pye ($34) or an Audiosonic ($30) will do. I have used a Pye for about 12 months without any trouble. The Audiosonic appears to contain exactly the same mechanism as the Pye, with the added benefit of a counter.

"Good condition" means that the recorder motor will keep a constant and correct speed, the record/playback head is properly aligned, and the machine is free of other mechanical or electrical defects. "Appropriate cable" means a cable wired to suit the DIN socket of the CoCo and the AUX, REM and MIC sockets of the recorder. Note that the AUX socket emits the signal for the CoCo while the MIC socket receives the signal from the CoCo. The REM socket permits the CoCo to control when the recorder motor will run.

Assuming that you have such a system set up, I'll talk about using it.

LOADING PROGRAMS

If you have a tape which is supposed to contain a program, then the following instructions should load the program.

(1) Rewind the tape to it's beginning.

(2) Set the volume control to a suitable level.

(3) Lock down the recorder PLAY key. Note that if the REM plug is in positon the recorder cannot play the tape until the CoCo permits it.

(4) Type a loading instruction to the CoCo. If the recorded program is written in BASIC the instruction will use CLOAD, otherwise it will use CLOADM, thus: type CLOAD "NAME <ENTER> or CLOADM "NAME <ENTER>. "NAME" is an optional title of the program, to a maximum length of eight letters. Normally a name is used to identify and select a specific program among a number of different programs on the same tape. In the event that only program or repeats of the same program are on a tape or if the first program on a tape is to be loaded, then the name can be omitted. If all is well, an S will appear, followed in due course when the program is found, by a flashing F. When loading is complete the F will stop flashing and be followed by an OK. Type RUN <enter> and you're away!

ERROR MESSAGES.

If all is not well, you will get an error message or the CoCo will continue to search for a program until you press RESET. The most probable error message would be I/O ERROR. Some of the common causes of I/O ERROR are:

(1) Starting to load a program in the middle.

(2) A recorder output level set too high or too low (probably too low, so that the CoCo fails to "hear" the recording properly).
(3) A faulty tape, resulting in an incorrect sequence of numbers reaching the CoCo. The fault may be due to such factors as wear, physical damage, stretching, heat, and moisture.
(4) Poor recording, in consequence of a dirty head, head misalignment, inadequate prior erasure (use a bulk eraser), etc.
(5) Deterioration of a recording by heat or electromagnetic radiation.
(6) Poor playback, for the same reasons as in (4).
(7) A broken lead, making contact most of the time.
(8) Using battery power, especially with weak batteries.

The CoCo responds with I/O ERROR when it's count of the data from the tape differs from a value written with the data when it was recorded.

Another error message which may occur is FM ERROR. This will result from an attempted CLOAD of a machine language program, or a CLOADM of a BASIC program. There are other reasons for this error message which are outside the scope of this article.

If your CoCo continues to show an S, you may check if any of the following are the cause:
(1) Recorder volume at minimum.
(2) AUX plug plugged into MIC socket and vice-versa.
(3) Blank tape?
(4) Program recorded while the CoCo was in high speed mode, due to a prior POKE65495,0; (note that speeding your CoCo up to match will get you an I/O ERROR!).
(5) Wrong or misspelt name.

RECORDER AND CASSETTE CARE.

It must be clear from the discussion so far that some tender loving care of your equipment (tapes included) will probably reduce the incidence of I/O ERRORS to a minimum. I should add that a better understanding of your system and some technical knowledge will also help. Therefore, let me give you a short chat about maintenance of your recorder and cassettes.

THE RECORDER.

Keep the exterior of the machine clean. Periodically wipe your recorder and it's cables with a damp cloth. Wipe dry with a clean, lintless cloth. If you live in a dusty area, protect the recorder with a dust-cover when not in use.

Inspect the cables connecting the recorder to the CoCo. Normally no wear will occur; but, should they chafe, replace them (or repair them yourself; it's not difficult).

Clean the recorder head using cotton buds and Freon or pure alcohol. Do not use domestic methylated spirits; it contains non-volatile components which will remain on the head after cleaning, thereby interfering with record and replay, and possibly contaminating your tapes. I suggest the head cleaning kit sold by Tandy or similar. Whatever you buy, avoid the abrasive cassette variety.

When operating the recorder, press the STOP/EJECT key in-between using other function keys.

Use mains rather than battery operation.

TAPE CARE.

Do not use your masters routinely. Make and use back up copies of all masters, including commercial masters if possible. Wrap your masters in aluminium foil and keep them in a dry place; away from electromagnetic radiators such as the Hi Fi, speakers, ham radios, large electric motors, transformers etc.

Keep all tapes in cases, ideally in cassette cabinets. this prevents dust pick up and consequent abrasion of tapes and recorder head.

Find the time to PLAY your tapes through on a regular basis, and especially before storage if you have done a FAST FORWARD or REWIND, thus avoiding tape stretch.

Do not touch the tape itself. any contamination of a tape by fingers etc can find its way throughout your range of cassettes, masters included.

Keep your bulk eraser well away from tapes that you do not want erased. (Like in the garden shed - G.)

MAKING BACKUP COPIES.

I mentioned earlier that you should avoid the routine use of your masters.

Whether or not you are making working copies or copying during program development, you will require the same setup as used for loading programs. Additionally,

1. Ensure that you are not going to to copy onto the leader, (the colored stuff at the start of the tape).
2. Have both RECORD and PLAY keys locked down.

Now type an instruction to CoCo to save your program. If the program is in Basic then the instruction will be CSAVE"NAME" (ENTER). As before, "NAME" can be omitted. I strongly suggest that you make two copies to reduce the risk of program loss due to I/O ERROR, especially while you are unaware of the level of reliability of your system. You can check whether your saved programs are OK or not by rewinding your tape to the start, and typing SKIPF"X" (ENTER). If your recordings are sound, CoCo will play them through. Otherwise an I/O ERROR is shown. At this stage you can resave until you get it right because the program is still in memory.

Saving machine language programs is a little more involved. The instruction to CoCo is of the form: CSAVE"NAME", start address, end address, execute address. To get those addresses, do the following:

For the start address type:
PRINTPEEK(487)*256+PEEK(488) (ENTER)
For the end address type:
PRINTPEEK(126)*256+PEEK(127)-1 (ENTER)
For the exec address type:
PRINTPEEK(157)*256+PEEK(158) (ENTER)

Insertion of those addresses into the command will do the trick!

If any of this article has confused you, please feel free to ring me on 07 208 0893 to discuss it further.

SCREEN PRINT

by Gordon Thurston.

Have you ever wanted to use graphics in a BASIC to printer program and found it all but impossible to implement? Well here is a program to solve that problem. It will convert a graphic screen in PMODE 0 to a BASIC program which it can then send to the printer.

The BASIC program can be edited normally and the screen print to the printer is very fast. One BASIC screen dump I tried took 1.5 hours to dump. This one completes the dump in a matter of seconds. The graphics can also be incorporated into lines of text making it ideal for letter heads etc.

I have not set the ORG in the listing so you will have to add a line 90 to set the start address ORG of the assembled program. I recommend for 16K machines 16000 and for 32K machines 32383. If you are POKEing the listing in from BASIC begin POKEing your data at either of those two addresses.

Before loading the assembled MACHINE LANGUAGE program into memory you must reserve some memory using CLEAR 16000,200 (16K) or CLEAR 32383,200 (32K). Once you have loaded the program you may then load any PMODE0 graphics program or alternatively draw your own screen. Once you have a picture you wish to dump to the printer, press (BREAK) and type NEW (ENTER). Now EXEC 16000 (EXEC 32383 for 32K) and after the OK prompt returns (about half a second), LIST will show that you have a BASIC program. Edit as you wish. You may RUN it as soon as you are ready.

This program works very well on my DMP110 printer and cassette based 16K CoCo 2. The program should work with most printers after making appropriate changes to the printer variables in the machine language listing.

Listing 2 is a sample program which draws a picture of a Piano keyboard in PMODE0. You can use this to test the workings of the screen dump program.

Listing 1:

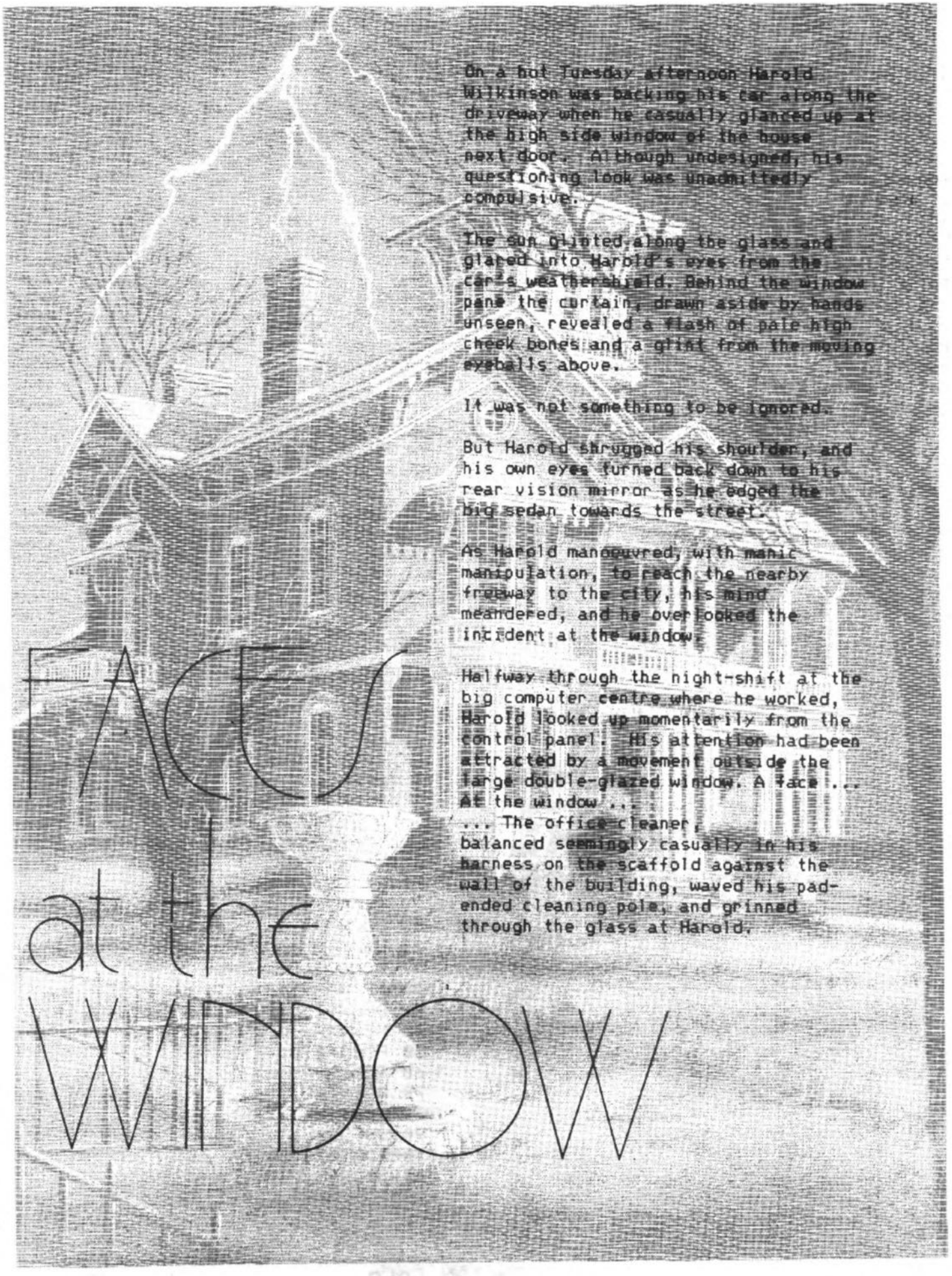
```
                    00100 ****PMODE0 PRINTER****
0000 8E   0600      00120 START   LDX     #$600   BEGIN PAG
E 1
0003 AF   8D 00B8   00130         STX     SCREEN,PCR      S
AVE IT
0007 86   0E        00140         LDA     #14     LENGTH OF
 LINE
0009 A7   8D 00B4   00150         STA     LINE,PCR        S
AVE IT
000D 86   08        00160         LDA     #8      8 BITS
000F A7   8D 00A9   00170         STA     COUNR,PCR       S
AVE IT
0013 86   80        00180         LDA     #128    SET BIT 7
0015 A7   8D 00A4   00190         STA     ROTAT,PCR       S
AVE IT
0019 86   10        00200         LDA     #16     NUMBER OF
 LINES
001B A7   8D 009C   00210         STA     COUNT,PCR
001F C6   07        00220         LDB     #7      NUMBER OF
 ROTATES
0021 86   12        00230         LDA     #18     SET PRINT
ER TO GRAPHICS MODE
0023 17   009C      00240         LBSR    BASIC   PUT INTO
BASIC PROGRAM
0026 86   1B        00250         LDA     #27     ESCAPE CO
DE
0028 17   0097      00260         LBSR    BASIC
002B 86   0E        00270         LDA     #14     DOUBLE WI
DTH MODE TO PRINTER
002D 17   0092      00280         LBSR    BASIC
0030 6F   8D 008A   00290         CLR     STORE,PCR       U
SED TO BUILD 1 BYTE TO PRINTER
0034 DE   19        00300         LDU     25      BEGBAS
0036 33   42        00310         LEAU    2,U     ADD 2
0038 DF   1B        00320         STU     27      ENDBASIC
003A A6   84        00330 LOOP    LDA     ,X      FIRST BIT
003C A4   8D 007D   00340         ANDA    ROTAT,PCR       M
ASK BIT
0040 27   0A        00350         BEQ     NONE    BYTE FORM
ED?
0042 86   80        00360         LDA     #128    SET BIT 7
0044 AA   8D 0076   00370         ORA     STORE,PCR       A
DD TO STORE
0048 A7   8D 0072   00380         STA     STORE,PCR       S
AVE BITS
004C 30   88 10     00390 NONE    LEAX    16,X    INITIATE
NEXT BIT TO PRINTER
004F 1C   FE        00400         ANDCC   #$FE            M
ASK Z BIT
0051 66   8D 0069   00410         ROR     STORE,PCR       P
OS FOR NEXT BIT
0055 5A             00420         DECB            READY NEX
T BIT
0056 26   E2        00430         BNE     LOOP    GET REST
OF BYTE
0058 C6   07        00440         LDB     #7      INITIATE
NEXT BYTE
005A A6   8D 0060   00450         LDA     STORE,PCR       B
YTE FOR PRINTER
005E 8A   80        00460         ORA     #128    SET BIT7
0060 8D   60        00470         BSR     BASIC   SAVE
0062 6F   8D 0058   00480         CLR     STORE,PCR       M
AKE ROOM
0066 66   8D 0053   00490         ROR     ROTAT,PCR       C
HANGE MASK
006A AE   8D 0051   00500         LDX     SCREEN,PCR      N
EW BYTE
006E 6A   8D 004A   00510         DEC     COUNR,PCR       C
OUNT ROTATES
0072 27   02        00520         BEQ     NEX16   BYTE DONE
?
0074 20   C4        00530         BRA     LOOP    NO GO AGA
IN
0076 AE   8D 0045   00540 NEX16   LDX     SCREEN,PCR      N
EXT LINE
007A 30   01        00550         LEAX    1,X     POINT TO
NEW BYTE
007C AF   8D 003F   00560         STX     SCREEN,PCR      S
AVE IT
0080 66   8D 0039   00570         ROR     ROTAT,PCR       R
OTATE MASK
00B4 AE   8D 0037   00580         LDX     SCREEN,PCR
0088 B6   08        00590         LDA     #8
008A A7   8D 002E   00600         STA     COUNR,PCR       I
NITIATE NEXT ROTATE
008E 86   80        00610         LDA     #128
0090 A7   8D 0029   00620         STA     ROTAT,PCR
0094 6A   8D 0023   00630         DEC     COUNT,PCR
0098 27   02        00640         BEQ     NEWLN   LINE FINI
SHED?
009A 20   9E        00650         BRA     LOOP    NO GO AGA
IN
009C 86   0D        00660 NEWLN   LDA     #13     INITIATE
NEW LINE
009E 8D   22        00670         BSR     BASIC
00A0 B6   10        00680         LDA     #16
00A2 A7   8D 0015   00690         STA     COUNT,PCR
00A6 AE   8D 0015   00700         LDX     SCREEN,PCR
00AA 30   88 60     00710         LEAX    96,X
00AD AF   8D 000E   00720         STX     SCREEN,PCR
00B1 6A   8D 000C   00730         DEC     LINE,PCR
00B5 27   03        00740         BEQ     DONE    LAST LINE
?
00B7 16   FF80      00750         LBRA    LOOP    NO AGAIN
00BA 39             00760 DONE    RTS
00BB      00        00770 COUNT   FCB     0
00BC      00        00780 COUNR   FCB     0
00BD      00        00790 ROTAT   FCB     0
00BE      00        00800 STORE   FCB     0
00BF      0400      00810 SCREEN  FDB     0
00C1      0E        00820 LINE    FCB     14
00C2 6D   8D 005A   00830 BASIC   TST     FLAG,PCR        L
INE UNFINISHED?
00C6 27   10        00840         BEQ     INITIL  NO? START
 ONE
00C8 10AC 8D 0009   00850         CMPY    ENFLAG,PCR      L
INE LONG?
00CD 27   56        00860         BEQ     ENLINE  YES? FINI
SH
```

```
00CF 81   0D       00870          CMPA    #13      END OF LI
NE?
00D1 27   52       00880          BEQ     ENLINE   YES? FINI
SH LINE
00D3 A7   A0       00890          STA     ,Y+      WRITE BYT
E TO BASIC
00D5 39            00900          RTS
00D6      0000     00910 ENFLAG   FDB     0
00D8 109E 1B       00920 INITIL   LDY     27       START NEW
 LINE IN BASIC
00DB 31   3E       00930          LEAY    -2,Y
00DD 10AF 8D 003F  00940          STY     NEXLN,PCR      S
TORE ADDRESS OF LINE ADDRESS
00E2 31   A9 008E  00950          LEAY    142,Y
00E6 10AF 8D FFEB  00960          STY     ENFLAG,PCR     F
LAG END OF LINE
00EB 31   A9 FF74  00970          LEAY    -140,Y   START ADD
RESS OF LINE
00EF EE   8D 0030  00980          LDU     LINENO,PCR     G
ET CURRENT LINE NO.
00F3 EF   A1       00990          STU     ,Y++     STORE IT
00F5 33   4A       01000          LEAU    10,U     NEW LINE
NO.
00F7 EF   8D 0028  01010          STU     LINENO,PCR     S
AVE IT
00FB 34   02       01020          PSHS    A        SAVE GRAP
HIC BYTE
00FD 86   87       01030          LDA     #135     TOKEN FOR
 PRINT STATEMENT
00FF A7   A0       01040          STA     ,Y+
0101 86   23       01050          LDA     #35      CROSSHATC
H
0103 A7   A0       01060          STA     ,Y+
0105 86   AC       01070          LDA     #172     TOKEN FOR
 MINUS SIGN
0107 A7   A0       01080          STA     ,Y+
0109 86   32       01090          LDA     #50      NUMBER TW
O
010B A7   A0       01100          STA     ,Y+
010D 86   2C       01110          LDA     #44      COMMA
010F A7   A0       01120          STA     ,Y+
0111 86   22       01130          LDA     #34      QUOTATION
 MARK
0113 A7   A0       01140          STA     ,Y+
0115 86   01       01150          LDA     #1       FLAG FOR
LINE STARTED
0117 A7   8D 0005  01160          STA     FLAG,PCR
011B 35   02       01170          PULS    A
011D A7   A0       01180          STA     ,Y+      GRAPHIC B
YTE TO BASIC LINE
011F 39            01190          RTS
0120      00       01200 FLAG     FCB     0
0121      0000     01210 NEXLN    FDB     0
0123      03E8     01220 LINENO   FDB     1000
0125 A7   A0       01230 ENLINE   STA     ,Y+      FINISH BA
SIC LINE
0127 34   02       01240          PSHS    A
0129 86   22       01250          LDA     #34      QUOTATION
 MARK
012B A7   A0       01260          STA     ,Y+
012D B6   3B       01270          LDA     #59      SEMI COLO
N
012F A7   A0       01280          STA     ,Y+
0131 B6   00       01290          LDA     #0       MARKS END
 OF BASIC LINE
0133 A7   A0       01300          STA     ,Y+
0135 EE   8D FFE8  01310          LDU     NEXLN,PCR
0139 10AF C4       01320          STY     ,U       STORE POI
NTER TO NEXT LINE
013C A7   A0       01330          STA     ,Y+
013E A7   A0       01340          STA     ,Y+      LAST 2 ZE
ROS AT END OF BASIC PROG.
0140 109F 1B       01350          STY     27       ENDBAS
0143 A7   8D FFD9  01360          STA     FLAG,PCR
0147 35   02       01370          PULS    A
0149 39            01380          RTS
           0000    01390          END
00000 TOTAL ERRORS

BASIC   00C2
COUNR   00BC              LOOP    003A
COUNT   00BB              NEWLN   009C
DONE    00BA              NEX16   0076
ENFLAG  00D6              NEXLN   0121
ENLINE  0125              NONE    004C
FLAG    0120              ROTAT   00BD
INITIL  00DB              SCREEN  008F
LINE    00C1              START   0000
LINENO  0123              STORE   008E
```

Listing 2:

```
  1 CLEAR1000
  5 A$="D60R10U20L3U40L7R10"
  6 D$="L3D40R6U40L3"
  7 E$="R3D40L3D20R10U20L3U40L4R7"
  8 F$="R10D60L10U20R3U40R7"
 10 PMODE0,1
 20 SCREEN1,1
 30 PCLS
 35 DRAW"BM10,10
 37 FORA=1TO3
 40 DRAWA$+D$+E$+D$+F$
 42 DRAWA$+D$+E$+D$+E$+D$+F$
 45 NEXT
 50 FORA=0TO14
 60 READB
 70 PAINT(B,15),1,1
 75 NEXT
 90 GOTO90
100 DATA20,30,50,60,70,90,100,12
0,130,140,160,170
105 DATA190,200,210
```



FACE at the WINDOW

On a hot Tuesday afternoon Harold Wilkinson was backing his car along the driveway when he casually glanced up at the high side window of the house next door. Although undesigned, his questioning look was unashamedly compulsive.

The sun glinted along the glass and glared into Harold's eyes from the car's weathershield. Behind the window pane the curtain, drawn aside by hands unseen, revealed a flash of pale high cheek bones and a glint from the moving eyeballs above.

It was not something to be ignored.

But Harold shrugged his shoulder, and his own eyes turned back down to his rear vision mirror as he edged the big sedan towards the street.

As Harold manoeuvred, with manic manipulation, to reach the nearby freeway to the city, his mind meandered, and he overlooked the incident at the window.

Halfway through the night-shift at the big computer centre where he worked, Harold looked up momentarily from the control panel. His attention had been attracted by a movement outside the large double-glazed window. A face ... At the window ...
... The office cleaner, balanced seemingly casually in his harness on the scaffold against the wall of the building, waved his pad-ended cleaning pole, and grinned through the glass at Harold.

The man at the computer, engaged in his workplace, his hands unfree, nodded upwards, and with a sly, unwelcome smirk, went back to his programming.

It was not until the early hours of the following morning that Harold Wilkinson recalled the fleeting incident of the evening before, just as he had left his home. The face at the window. Or two faces? Questions needed answers. There should be no-one there. The house next door, being sold, was meantime vacant. Harold knew that the former owners,the Newlings, a retired couple, had disposed of the house fully furnished, and had left by 'plane at the weekend for an overseas tour. Had it really been a face? Or was it just imagination? Or an optical illusion caused by the glaring summer sun and a slight draught making the curtains move just as Harold had left for work the previous afternoon? What was in his mind? Something implanted?

Again Harold looked up - almost with an air of anxiety - as he nudged the car towards his garage. ... Nothing. ... A blank, blacked-out, flat square, moistened by the heavy night dew and shadowed by the overhang of the eave in the morning's half-light.

However, welcomed by his tailwagging, trembling terrier, Harold quickly tossed off the cares of the day (or should we say - night?), and with them any thought of the face he had seen, or maybe thought he had seen, at the window next door. Resultantly, there was no mention of the incident at the family breakfast table.

The next afternoon, Harold rear-ended his big green sedan automatically along the side of his house without even a glance sideways or upwards, his mind already programming his car's route against the oncoming traffic as he prepared to make his hasty way citywards.

Halfway through the night's work, whilst keying a statistical upwards market curve, Harold's eyes mechanically followed the thin lines on the screen in front of him, and his mind, computerlike, repeated the pictures of the day before.

Faces at the window ...Fleeting faces, not unlike those he'd seen his kids peering at, playing at their home video games.

Harold looked up at the office wall and its blank uncurtained glazing sheet ... Nothing. Except the mirrored waves of reflections from the lines of fluorescent tubed lighting glaring down and sideways from the pinholed ceiling surfaces. Harold's eyes returned first to the panel in front of him, then back to his keyboard. His hands obeyed the impulses from his eyes and his brain, and like a maestro at an organ, Harold went on with the evening's programming, his eyes flitting amongst the figures and sketches on the pads alongside him, and his fingers tapping in type-touching rhythm. The night's work went well, and as Harold turned out the lights and power and watched the fading glow of the switched-off videos, he got a feedback message of achievement ... mission accomplished ... problem solved.

But a few minutes later, as he unlocked his car door in the gloom of the cavernous parking site beneath the computer centre, Harold's mind projected into the shadowed flatness of the vehicle's window glass, another picture - the face, the high-cheeked face he'd seen at the window next door to his home. Meandering imagination? ... A mindless message from his overtaxed brain. Or was there a hidden meaning that should impel him into some sort of action?

Harold was still pondering the inexplicable when he reached home. Too tired to worry, he was at the point of thrusting the vague wearying unease right out of his mind, when he glanced up uncertainly, involuntarily, at the window beside his driveway.

What he saw then propelled Harold's thoughts, his eyes, his hands, his feet. He squealed his car to a grunting, shuddering stop, the gears growling in protest as he forgot to disengage the lever. ... A face at the window! ...Again ... Or had it never left that spot ...?

Continued on P 83

DISK ZAP

by Barry Daraton

Everyone that owns a disk knows the feeling when it crashes. Oh well out with the backup and re-copy it. Thats if you've got a complete backup and you also know what was on that particular disk.

Its always a good idea to have a backup copy of each disk on hand but after a while you get quite a large accumulation of them and keeping backup copies of seventy or more disks can be terribly expensive especially if you run single sided drives.

I always used to keep a backup of every program I had until one day I got a program which I decided to play with before saving a backup copy, and you guessed it, it zapped on me in a big way, and took all the other programs on the disk with it, and with no backup on hand (besides the fact that I did'nt know what was on there because DIR gave a garbage display on the screen).

Well out with the zapper program I bought about six months ago and destroyed more disks than I salvaged. There was a great book of instructions with it which I spent about three weeks reading to understand about 1 percent of. I think that the people that write these manuals assume that you are a fully fledged computer genius, which is definitely not me.

What I intend to do here is to try to explain my method of repairs in a language most people would understand, keeping in mind that I am only writing from personal experience and reading documentaries on disk structure (well thats my excuse for any mistakes that I make).

A few things you will need to know to understand how to repair zapped disks are.

1 TRACKS.....a total of 35 or relative 34.
2 granule.....=1/2 a track.
3 sector.....18 sectors per track, a total of 630
4 byte.....1 byte= 1 number or character, 256 bytes per sector.

Did you spot that word RELATIVE (no not your mother in law), relative in this case means relative to. The first track on your disk is track 0 not track 1 so that means that relative track 0 is really track 1 or, track 35 is relative track 34, sounds confusing does'nt it. Think about it carefully because you will need to understand it properly to avoid getting lost later.

A granule is the smallest part of the disk that a program can be stored on. If you write a program that has only one word in it (eg. 1 REM) then this program would use one granule of the disk. so the maximum no. of programs that can be stored on a disk would be sixty eight.(PRINT FREE(0)) blank disk =68, 35 tracks, two granules to a track (assuming you are using a TANDY DOS) 35*2=70-2(directory track)=68

A sector is really explained about as much as you can above, there is more on a sector that you can't see or modify (with most zappers) called identification bytes which tell the DOS where it is reading from or writing to, but these are not really of any use to us for repairing.

A byte is also self explanatory because one byte is the same as in the computer (8 bits), eg one character (A)=61 (hex).

The tool required for repairing disks is a zapping program preferably a M/L type for speed purposes, a basic one is ok but a bit slow. The first thing to do is to familiarize yourself with whichever program you are using, I suggest that you practise on an experimental disk first just in case.

GETTING STARTED

The most important track on the disk is track seventeen,this track is divided

into two main sections.

1.....The directory sectors 3 to 11

2.....the granule allocation table sector 2.

Sectors 12 to 18 are not used at present by TANDY DOS nor can programs be stored on these sectors. Each entry (filename) on the directory track is allocated 16 bytes, (the other 16 are not used)

Bytes 0 to 7 are for the filename
8 to 10 are for the extension (BIN BAS DAT etc)
11 is for the file type
12 is the file stored in ascii or bin (m/l)?
13 the No. of the first granule in the file (relative)
14 to 15 the No of bytes in the last sector (byte 14 will be zero)
16 to 32 are not used at this time by TANDY DOS.

```
00 01 02 03 04 05 06 07 08 09  10     11
12 13       14  15

00  01  02 03 04 05 06 07 08 09 0A    0B
0C 0D       0E  0F

C  O  M  P  U  T  E  R  B  I  N     02
00 20       00  CB

C  O  M  P  U  T  E  R  B  I  N   bin
track 16,1  203 bytes
              into last
                sector

C  O  M  P  U  T  E  R  B  A  S     00
00 28       00  F9

C  O  M  P  U  T  E  R  B  A  S   bas
track 21,0  249 bytes
              into last
                sector
```

I bet you're confused about the track no. how does the number 20 in byte 13 convert to track 16 sector 1.

First you have to convert the number you get from byte 13 to decimal, 20(hex)=32, now divide this No. by two, 32/2=16 and there you have it track 16. There is no remainder so it is the first sector. (first sector is 1 not 0).

But what about the the other one, byte 13 =28(hex) which=40(dec) ,now 40/2=20 not 21, or at least it did when I went to school, so why does this become track 21,1. Remember track 17 is the dir track and so it is not included in the disk as far as program storage goes, so if when you convert byte 13 to the track No. and this No is greater than 17 then you must add 1 to it to get the right track No. however if the No. is less then 17 then it is the right track No.

That was'nt so hard was it?, well neither is the next part.

```
 C  O  M  P  U  T  E  R  B  I   N     02
00 13       00  F3
```

Byte 13 =13(HEX) which=19(dec), 19/2=9 1/2. so what now?. Well we add 1/2 a track. 18 sectors per track half 18/2=9 +1 because sectors start at 1 not 0 so this is now track 9 sector 10.

I'll leave a few examples with you next. Make sure that you understand how to do this exactly without any errors or you will wind up loading chess and getting space invaders, if you're lucky enough to get anything.

byte 13 (dec)	add	+ 10	track	sec
divided by 2.	track	sect		
6 =6 =3	no	no	3	1
12=18=9	no	no	9	1
16=22=11	no	no	1	1
20=44=22	yes	no	23	1
43=67=33 1/2	yes	yes	34	10

You should be able to work it out for yourself now, if you can't then read the first section again. Be sure that you fully understand granule to track conversion before continuing.

Now that the easy part is firmly implanted in your brain we now go onto track 17 sector 2, the granule allocation table.

This is probably the hardest sector to understand and will take quite a bit of thinking to understand properly.

As explained before there are 256 bytes in a sector but only the first 68 (in this sector) are important to us, got it?, 68 bytes and 68 granules.

You guessed it each byte points to a granule, so if byte 30(hex) has any

other number than FF then there is something on that sector, lets look at a dir track and a GAT of a disk with one program on it.

```
C O M P U T E R B I N 02 00 1A 00 C7

     (hex)                G.A.T.
BYTE 00-10        FF FF FF FF FF FF FF FF
                  FF FF FF FF FF FF FF FF
BYTE 11-20        FF FF FF FF FF FF FF FF
                  FF FF 1B 1C 1D 1E C7 FF
BYTE 21-30        FF FF FF FF FF FF FF FF
                  FF FF FF FF FF FF FF FF
BYTE 31-40        FF FF FF FF FF FF FF FF
                  FF FF FF FF FF FF FF FF
```

Your display may vary to this one but the meaning is the same.

Now lets do some calculations:
First the track No. 1A(hex)=26 decimal, 26/2=13, 13 is less than 17 so 13,1 is the track and sector No, (sectors start at NO. 1).

Second we go to track 17 sector 2 (G.A.T) and look at byte 1A because thats where the dir track told us where to look. We find at byte 1A the numbers 1B 1C 1D 1E C7, this tells us how many granules (1/2 tracks) there are in the program and where on the disk we find them. The value of each byte in the G.A.T is a pointer to the location of the next granule used by the file stored on the disk. The last granule used by a file contains the hex value C as its first character.Let's look at it in more detail.

```
 Starting point =13,1
 No of granules = 5    1A 1B 1C 1D 1E
```

Add 5 granules to track 13, 2 granules to a sector 5/2=2.5, so track 13,1+2.5 sectors= 15,9. (1/2 a sector is 9 tracks)

Now the last No in the G.A.T is a C7, this points to the last sector in the last granule, now it sounds confusing does'nt it.

So far we know that the program goes to 15,9 the C7 tells the DOS that the last sector is number 7. So now we know the last address of the program on the disk is track 13 sector 7.

Now I suppose that you are wondering why the numbers in the G.A.T point to sector 9 then the next No. points to sector 7, It's not really that hard to understand if you think about what was said earlier. 1 granule is the smallest portion of a disk that can be used for storage, so the last No. in the G.A.T must contain a No. other than FF or the DOS does'nt know that there is a program on that section and can overwrite it when a new program is saved to it. 1 granule=9 sectors, the C7 tells DOS to only read the first seven sectors of this granule.

To find the last byte in the program we must go back to the directory track, (17,3) and look at the No. in the 15 byte which in this case is also C7(no they dont always match). If you go to this byte in the last sector you will know exactly where your program ends.
We'll do one more example then go onto programs bieng stored on the disk in a backward fashion rather than forward.

```
C O M P U T E R B I N 02 00 30 00 5A
                 G.A.T
 31 32 32 34 35 36 37 C4
 Starting    track     =30(hex)=48(dec);
48/2=24,  higher than 17 so add 1,1t now
is 25,1
 No of granules = 7, 7/2=3.5 tracks
Actual start =25,1 end=track 28
 Add No of sectors in  last  granule  =4
(C4)
 total  st=25,1:  end=28,4: last byte is
5A (90)dec
```

I'll do one more using the way that I scribble down on my notepad then I'll leave 1 for you to do.

```
C O M P U T E R B I N 02 00 3A 00 BF
                 G.A.T=3B 3C 3D 3E 3F C4
 3A=58/2=29+1=30:5   GRANULES/2=2     1/2
TRACKS:   30+2=32+9   sectors   (1/2   a
TRACK)+4 SECTORS  (C4):  TOTAL=TRACK  32
SECTOR 13.
HERES ONE FOR YOU.
C O M P U T E R B I N 02 00 12 00 BD
  G.A.T.=13 14 15 16 17 18 19 1A C4
```

All this is ok as long as the numbers run in ascending order, what when they go back to front. Well all you've got to do is to subtract instead of add.

```
C O M P U T E R B I N 02 00 1A 00 BD
  G.A.T.=1B 18 19 16 17 14 15 12 C7
```

1A=26/2=13: starting track 13 sector 1.

Add the number of granules = 8 (1B 18 19 16 17 14 15 12)
8 sectors /2=4 tracks: 13-4=9: add No of sectors in last byte of of the G.A.T.=7 (C7).so the total is now. Start 13,1 to 9,7.
1 more G.A.T.=1B 18 19 16 17 14 15 C7
1A=26/2=13:TRACKS=(7/2),31/2=13-3=10+1/2 track=10,9+7(C7)=10,16, total= start 13,1 to 10,16.

There is only one more to go, when your disk gets pretty well full up and you save another program, it may be stored all over the disk in any spare granule that is available. This can get real complicated so think real hard.

Byte 13 dir track = 1A
G.A.T = 1B 12 13 04 05 34 C7
The order of movement in this table will be as follows, start track=13,1to 9,1:04,1:27,1.
First convert all numbers in the G.A.T. to track numbers.

1A=26/2= 13 SECTOR 1
1B=27/2= 13 SECTOR 10
12=18/2= 09 SECTOR 1
13=19/2= 09 SECTOR 10
04=04/2= 02 SECTOR 1
05=05/2= 02 SECTOR 10
34=52/2= 27 SECTOR 1

Right these numbers tell the DOS what to do and how far to go

NO	from	to
1A	13,1	13,9
1B	13,10	13,18
12	9,1	9,9
13	9,10	9,18
04	2,1	2,9
05	2,10	2,18
34	27,1	27,9

The C7 tells the DOS to only read the first seven sectors of the last granule.

This next example is going to move the head to tracks in this order 13,1,34,2,34. Boy would'nt 6 milliseconds step be handy here.

Byte 13 dir track=1A
G.A.T = 02 43 04 42

1A=26/2=13 SECTOR 01
02=02/2=01 SECTOR 01
43=67/2=34 SECTOR 10
04=04/2=02 SECTOR 01
42=66/2=34 SECTOR 01

NO	FROM	TO
1A	13,1	13,9
02	01,1	01,9
43	34,9	34,18
04	02,1	02,9
42	34,1	34,9

C7 tells DOS to read only the first seven sectors of the 34th track.

Just for the practise get a blank formatted disk and write these values onto the disk using your zapper program. If your zapper has a function to step through the tracks and sectors of the program you name, then use it with these examples and then try different numbers untill you feel you know it pretty well.

RECOVERING A KILLED FILE

The next step is pretty easy. Using a blank formatted disk, get a M/L program that you know the addresses for and copy it on to the experimental disk. Next use the zapper to look up the info that you need on G.A.T and the directory track and then, kill the file off.

Next step is to look up track 17 sector 3, You will notice that only the first character of the filename is missing.
- O M P U T E R B I N 02 00 1A 00 5A
But the G.A.T. track is the problem, all the numbers determining the allocation of the program are missing. In this case it's easy to restore because you have the numbers to replace the ones that are missing, so 1 minutes work and you will be rolling again.By putting the numbers back on the G.A.T and then the missing character on the filename track. Depending on which zapper you are using, you should do the G.A.T first as one zapper I have tends to give a CRC error if I write to the filename sector first.

A CRC error by the way is pretty hard to explain in easy to understand words. (probably because I dont understand it either) but I'll give it a go anyway.

A checksum is in this case two sets of numbers on the disk,(one set by formatting the other set when writing)if the checksum does not match (when verifying) then we get a CRC error. Sometimes rewriting to the sector straight away after a CRC error will fix it, only sometimes. There is no real way

of finding the cause of a CRC error unless a bad sector is on the disk. If this is the case then use the verify disk command on your zapper if it has one to find the bad sector, or if no verify is available then you will have to step through the sectors one at a time until you can't read a sector, if you encounter no errors then your guess is as good as mine, it may have even been a temporary hardware hickup, loose controller or bad cable connections.

If you do encounter a bad sector in your verify, then it's easy solved. Convert the track and sector number into a granule no then go to the G.A.T, and write a C0 to the corresponding granule no. That will stop you rewriting to that granule, but don't hold up too much hope for the life expectancy of the disk, I'd transfer everything off that disk to another disk before it hangs up permanently. By the way once you deactivate this granule the basic backup command won't work, because it copies sector for sector, so when it hits the bad one, instant hangup. The two options open to you are to copy 1 program at a time from basic or use the copy sectors function of your zapper, and use it to miss the crook granule and get all the rest.

Well getting back to our killed disk which you should now have recovered, go to the first sector of the program (thats the one that byte 13 points to) and look at it in the hex mode, if you look at the first half a dozen bytes or so then you will see the start address of the program. To know for sure that you have the proper end address then go to the file name sector and get the number from the 15th byte and then go to the last sector of the file and then to the byte number that you just got from the filename sector and you will notice that the numbers preceeding that number will point to the execution address of the program. This can come in very handy for locating the last sector of a program (only M/L) on a disk that has really screwed up and written across the filename tracks.

REPAIRING A BADLY CRASHED DISK CAN TAKE UP TO 4 HOURS SO DON'T BE TO DISAPPOINTED IF IT TAKES YOU A WHILE TO GET USED TO IT.

Well I reckon thats about enough for now, we could go into fixes for a crashed basic program if you want, but I'll leave that up to you. If I get the message that you want more on this subject then I'll do more else I'll just leave you with this much to play with.

---

bass loader

16K ECB

I have always found it so frustrating to come across an interesting program in a magazine only to discover the listing is that dreaded Assembly Language. I know there have been other BASIC programs that POKE the data into memory but never having found one that does all that I want I finally wrote my own BASSLOADER.

Assembly Language programs usually appear with the Machine Code included with the listing and it is this we POKE into memory. I have included complete instructions within the listing and there is no need to repeat them here. (I hate having to hunt through my pile of junk to find the instructions for a program when it can easily be included in the listing).

Once you have keyed in the data simply typing XX will exit the input routine and prompt you through the program saving routine. You may save to either Cassette or Disk.

What more do you need. I even reckon that BASSLOADER is easier and quicker to use than an assembler. At least now you won't have to leave those assembly programs to the Guru's.

The Listing:

```
10 CLEAR200,32353
20 FORR=32353 TO 32381:READ P:PO
KE R,P:NEXTR
```

```
30 DEFUSR0=32353
40 DATA 189,179,237,31,152,142,4
,0,16,142
50 DATA 4,31,237,137,1,224,237,1
64,49,168
60 DATA 32,237,129,140,4,32,38,2
40,57
70 CLS:BN=234
80 CLS0:X=USR0(BN)
90 PRINT@135,"BASLOADER PROGRAM"
;
100 PRINT@197,"BY MICHAEL J. CAR
TER ";
110 PRINT@264,"8 CANBERRA ST.,";
120 PRINT@296,"TOOWOOMBA 4350.";
130 PRINT@328,"  QUEENSLAND.  ";
140 PRINT@360,"   AUSTRALIA   ";
150 PRINT@423," TEL. 076-35 6911
 ";
160 PRINT@450," [ HIT ANY KEY TO
 CONTINUE ]";
170 X=USR0(BN)
180 FOR R=1 TO I:NEXT
190 X=USR0(BN-3)
200 FOR T=1TOI:NEXT
210 IF I>1 THEN I=I-1 ELSE I=50
220 IF INKEY$=""THEN170
230 CLS:PRINT"**BASIC MACHINE LA
NGUAGE ENTRY**      BY: MICHAEL J
. CARTER.
240 PRINT:PRINT"THIS PROGRAM WIL
L ALLOW BASIC    USERS TO INPUT M
ACHINE LANGUAGE CODE FROM ML LIS
TINGS IN RAINBOWAND OTHER MAGS.
AFTER INPUT A    SAVE OR CSAVE WI
LL BE CARRIED    OUT.PROMPTS WILL
 GUIDE YOU       THROUGHOUT.
250 GOSUB 2000
260 CLS:PRINT"IN ML LISTINGS THE
 1ST. COLUMN  IS THE ADDRESS IN
MEMORY INTO   WHICH THE CODE IS
PUT. YOU WILL NOTICE THAT IF THR
EE BYTES OF   CODE ARE INPUT THE
 ADDRESS ON   THE NEXT LINE WILL
 BE           INCREMENTED BY THR
EE."
270 PRINT"example
          3F00 BE  0168   00100
START LDX
          3F03 AF  8D0021 00110
      STX
          3F07 86  7E     00120
      LDA"

280 PRINT"WE ARE ONLY INTERESTED
 IN THE   FIRST THREE COLUMN'S.
290 GOSUB 1000.
300 CLS:PRINT"KEY IN:- BYTE BY B
YTE THE TWO   CHARACTER HEX CODE
 IN SEQUENCE. STARTING WITH THE
 FIRST        INSTRUCTION FOLLOW
ED BY THE     DATA.
              CONTINUE UNTIL ALL
 CODE IS      ENTERED."
310 PRINT"example
          BE<ENTER> 01<ENTER> 68
<ENTER>   AF<ENTER> D8<ENTER>

          CONTINUE TILL DONE.
          ENTER <XX> TO END INPU
T."
320 GOSUB 1000
330 CLS0:PRINT"PROGRAM RUNNING P
LEASE WAIT."
340 FOR T=1 TO 1000:NEXT
350 CLS:INPUT"IN HEXADECIMAL PLE
ASE.          STARTING ADDRESS "
;S$
360 I=VAL("&H"+S$)
370 PRINT"&H"HEX$(I);:INPUT"  VA
LUE";B$
380 IF B$="XX" THEN 410
390 POKE I,VAL("&H"+B$)
400 I=I+1:GOTO370
410 CLS:PRINT"SAVE TO CASSETTE <
C>             SAVE TO DISK     <
D>
420 Z$=INKEY$:IF Z$<>"C" AND Z$<
>"D" THEN 420
430 IF Z$="C" THEN N$="CASSETTE"
:CLS:INPUT"CSAVE NAME? 8 CHARS.M
AX.";IN$:GOTO 330
440 IF Z$="D" THEN N$="DISK":CLS
:INPUT"SAVE NAME? 8 CHARS.MAX.
       /BIN ASSUMED.";IN$
450 CLS:PRINT"READY ";N$
460 PRINT"HIT ANY KEY FOR SAVE"
470 Y$=INKEY$:IF Y$="" THEN 470
480 ST=VAL("&H"+S$):EN=I:EX=VAL(
"&H"+S$)
490 IF N$="CASSETTE" THEN 510
500 SAVEM IN$+"/BIN",ST,EN,EX:GO
TO 520
510 CSAVEM IN$,ST,EN,EX
520 END
1000 PRINT@486,"ANY KEY TO CONTI
NUE";
1010 Y$=INKEY$:IF Y$="" THEN 101
0
1020 RETURN
2000 PRINT@486,"<I> INS.   <C> C
ONT.";
2010 Y$=INKEY$:IF Y$<>"I" AND Y$
<>"C" THEN 2010
2020 IF Y$="C" THEN 330
2030 RETURN
```

Robb Webb

This program won't make paying your taxes any easier, but it can calculate how much tax you should pay for a given income. It can also work backwards to calculate gross income from a given tax figure or nett income. The program output is designed for gross incomes less than $1,000,000 per annum (if there is anyone who finds that income level limits their use of the program I'll start eating porridge for breakfast K.M.).

Upon running the program, you see a few lines of instructions, then comes the menu, which is self explanatory. The program is presently configured for the tax scales applicable from 1st November 1984, but the thresholds, rates and minimum tax payable within each bracket can easily be altered by changing the DATA lines at the beginning of the program. Do not alter line 60 however, it is used as a dummy DATA line to count the number of tax brackets.

To check your program is running correctly, the following table can be used as a comparison chart:

GROSS INCOME	TAX	NETT INCOME
$4595.00	$0.00	$4595.00
10000.00	1351.25	8648.75
15000.00	2726.25	12273.75
25000.00	6606.25	18393.75
30000.00	8946.25	21053.75
50000.00	20346.25	29653.75

The Listing:

```
10 DATA4595,.25,0
20 DATA12500,.3,1976.25
30 DATA19500,.46,4076.25
40 DATA28000,.48,7986.25
50 DATA35000,.6,11346.25
60 DATA,,
70 READT(X),R(X),M(X)
80 IFT(X)THENX=X+1:GOTO70
90 X=X-1:C$="       $$####,###.##
100 GOSUB150:GOTO180
110 PRINT@214,STRING$(10,45);
120 RETURN
130 PRINT@458,"PRESS ANY KEY
140 IFINKEY$=""THEN140
150 CLS:PRINT@12,"TAXATION
160 PRINT@44,STRING$(8,45):PRINT
170 RETURN
180 PRINT"THIS PROGRAM ALLOWS CA
LCULATION OF SIMPLE TAXATION HYP
OTHESES.  DO NOT USE COMMAS WHEN
 ENTERING FIGURES. CALCULATIONS
ARE VALID FOR GROSS ANNUAL INCOM
ES OF LESSTHAN $1,000,000.
190 GOSUB130:PRINT"WOULD YOU LIK
E TO CALCULATE?
200 PRINT:PRINT"1) THE AMOUNT OF
 TAX PAYABLE ON    A SPECIFIED g
ross INCOME.
210 PRINT"2) GROSS INCOME REQUIR
ED TO GIVE   A SPECIFIED net INC
OME.","3) GROSS INCOME REQUIRED
TO GIVE   A SPECIFIED AMOUNT OF
tax.220 PRINT"4) NONE.
230 PRINT@416,"SELECT: (1-4)
240 B$=INKEY$:B=VAL(B$)
250 IFB<1ORB>4THEN240
260 IFB=4THENEND
270 GOSUB150:ONB GOTO280,340,430
280 LINEINPUT"GROSS INCOME: $";G
$:G=VAL(G$):PRINT@110,USINGC$;G
290 IFG<=T(0)THENT=0:GOTO320
300 FORZ=1TOX:IFG>T(Z)THENNEXT
310 Z=Z-1:T=M(Z)+(G-T(Z))*R(Z)
320 PRINTUSING"NET INCOME:   "+C
$;G-T:GOSUB110
330 PRINTUSING"TAX PAYABLE:  "+C
$;T:GOTO190
340 LINEINPUT"NET INCOME: $";N$
350 N=VAL(N$)
360 PRINT@108,USING"  "+C$;N
370 IFN<=T(0)THENG=N:GOTO410
380 FORZ=1TOX
390 IFN>T(Z)-M(Z)THENNEXT
400 Z=Z-1:G=T(Z)+(N+M(Z)-T(Z))/(
1-R(Z))
410 PRINTUSING"TAX PAYABLE:  "+C
$;G-N;:GOSUB110
420 PRINTUSING"GROSS INCOME: "+C
$;G:GOTO190
430 LINEINPUT"TAX PAYABLE: $";T$
440 T=VAL(T$)
450 PRINT@109,USING" "+C$;T
460 IFT=0THENG=T(0):GOTO490
470 FORZ=0TOX:IFT>M(Z)THENNEXT
480 Z=Z-1:G=T(Z)+(T-M(Z))/R(Z)
490 PRINTUSING"NET INCOME:   "+C
$;G-T:GOSUB110
500 PRINTUSING"GROSS INCOME: "+C
$;G:GOTO190
```

RTTY

by Roy Lopez.

As a result of requests from the amatuers to both Graham Morphett and myself, for information on Radioteletype (RTTY) and Morse Code (CW), this article is being presented. While not making a claim to being an expert on either subject, neverthless I do have equipment 'up and running' which is generating and resolving both modes using CoCo, and in my search for success I have accumulated quite a deal of information on these modes of operation.

Right at the start it should be stressed that the transmission of any sort of information within the frequencies set aside for the use of Licenced Radio Amateurs, is legal ONLY for those who hold a CURRENT Amateur Radio Station Licence. Penalties for illegal or 'PIRATE' operation are severe, and in addition to confiscation of equipment may also result in a stiff fine and/or imprisonment. Information on obtaining an Amateur Radio Licence may be obtained from any of the State Branches of the Wireless Institute of Australia (WIA), or from a Regional Office of the Federal Dept. of Communications.

The aim of this article is to provide information at a level to suit those Radio Amateurs and others, who have little or no knowledge of these modes.

In order to successfully generate and transmit, receive and resolve computerised RTTY and CW, there must be suitable interfacing between the computer and the transmitting and receiving equipment. CW generated using a hand-operated key, or using a semi-automatic or automatic device; or RTTY generated from a teletype (TTY) keyboard, may also be received and resolved with varying degrees of success. Because CoCo and his mates set such exacting standards in regard to accuracy of speed and timing, the received RTTY or CW must be near perfect in order to be correctly resolved by the computer. Computer generated code, whether CW or RTTY, will match the exacting standard requirement, while mechanically generated code may not.

Even with computer to computer operation the problem of signal to noise ratio; atmosphere and other forms of interference; voltage variations and 'Spikes', all have to be coped with and indicate the necessity for adequate filtering in order to avoid misinterpretation by the receiving equipment and computer.

Commercial equipment is available, (at a price), to suit amateur requirments in this regard, however the fairly simple and relatively inexpensive interface to be described does an excellent job on both transmit and receive on my ECB COCO, converted to 32K (or 64K depending on your point of view), using 8 X 150/ns 6146 ICs and the removal of a few capacitors and the alteration of four latches.

For RTTY/CW operations on the Amatuer Bands you need:-

1. Amatuer Radio Station Licence.
2. Transmitting and receiving equipment.
3. Interface for CW. Modulator Demodulator (MODEM). Terminal unit (TU). Encoder Decoder; Black Box (call it what you will) for RTTY.
4. Computer with associated monitor or TV screen.
5. RTTY/CW program(s) to suit the computer in use.

For those who would like to try their hand at writing their own RTTY/CW program(s), the following information may be of assistance. Sophisticated programs have large buffers for both transmit and receive with the ability to DUMP to PRINTER, TAPE or DISC as required. Variable speed allowing CW speeds from 1-99 WPM and a range of BAUDOT and ASCII speeds to suit current standards, and character conversion lookup tables within the program. The use of a split screen displaying the received message on one half of the screen while 'KEYING IN' an answer on the other half is also useful. Canned messages e.g. the initial CQ call; QSL information, etc., add flexibility as

does the ability to switch from transmit to receive at the touch of a key.

EXAMPLES OF SUITABLE PROGRAMS are:-

1. 'HAMSHACK' for CW which appeared in 80-MICRO for December 1982 and the fixes appearing in later issues.
2. 'COLORFUL RTTY' which appeared in 73 magazine for September 1983.
3. 'NEWRYTCW' available from CLAY ABRAMS SOFTWARE, 1758 Comstock Lane, SAN JOSE, CALIFORNIA, 95124. Cost approx $30, specify to suit AUSTRALIAN MODEL COCO.

Like CW, RTTY uses a BINARY system. Where CW characters are made up of a series of dots and dashes, RTTY characters use marks and spaces. Amateur RTTY is transmitted using FREQUENCY SHIFT KEYING (FSK) with a centre frequency of 2210 HZ. The mark frequency is 2125 HZ and 170 HZ, shift has a space frequency of 2295 HZ while 850 HZ shift has 2975 HZ. Although the convention is that 850 HZ shift be used above 30 MHZ, 170-HZ shift appears to be in fairly general use by amateurs.

RTTY may use either the BAUDOT (MURRAY) or ASCII coding system. With the use of only 5-BITS, BAUDOT has only 31 possible character combinations, however with the use of 'LETTER' and 'FIGURE' shifts this is almost doubled. Because of it's limitation not all keyboard characters are available using BAUDOT. ASCII uses 7 BITS with 127 possible combinations and a shift also used making it much more flexible than BAUDOT.

Most amateur RTTY is transmitted using 45.45 BAUD (BITS per sec or BPS), also referred to as '60 speed' (Approx. 61.33 WPM), BAUDOT. Each BAUDOT character is made up of a start pulse 22 ms long, followed by 5 signal pulses each of 22 ms, followed by a stop pulse of 31 ms. Since pulse length in ms=1000/BAUD rate; 45.45 BAUD BAUDOT = 1000/45.45. The CoCo normal 600-BAUD RS-232 OUTPUT=1000/600=1.667-ms.

Before commencing the construction of the MODEM, I mounted a 5-PIN DIN socket above the channel switch at rear of the CoCo's chassis, and recessed the back of the top cover to accept the 5-PIN plug. The 5-PIN socket is parallel wired with the normal 4-PIN RS-232 socket, the fifth pin being connected to the +5V Joystick supply. However with the MODEM to be described only 3 of the RS-232 connections are needed, PIN 1 NOT being used. PIN 2 is INPUT; PIN 3 is chassis and PIN 4 is OUTPUT from the computer. I leave the DMP-100 printer connected to the 4 PIN RS-232 socket so that the buffers may be emptied as required, while the MODEM remains connected to the 5-PIN DIN socket.

The MODEM circuitry is based on a design by Michael J.DI. JULIO which appeared in QST for December 1980 page 20. My MODEM is mounted in an aluminium 'Home-Brewed' box 14cm square by 15cm deep with the controls arranged around the meter which is central to the the front panel. All sockets and S6 a slide switch, together with the shielded cables to P2 and P3, and the 12V power socket are mounted on the rear panel. The 9V DC power requirement is obtained by means of a suitable dropping resistor, and as I take my 12V from the transceiver supply, I included a seperate voltage regulator on the MODEM PCB.

No PIN numbers are shown on P1, J6, P2 or J7 as this will be determined by the mic and socket wiring of the transceiver in use. Mic plug P1 mates up with J6 and P2 with J7, while P3 mates with J8. Since the MODEM was originally designed for a TTY KEYBOARD/PRINTER operation. If this is not a requirement in your case, J3 and the 4 X in 4003 DIODES and S4B may be dispensed with. Since a TTY printer may have a loop voltage of 120V DC or more the computer should not be in circuit when J3 is in use. S6, Q7, Q8 and associated circuitry provide an interface to drive a TTY pprinter, from the computer via J5. If this is not a requirement in your case, then items S6 to J5 may be dispensed with.

J1 connects to the transceiver. EXT.SPKR./PHONES socket mutes the speaker voice coil. Since a plug in the EXT.SPKR./PHONES socket mutes the speaker, J1A takes a speaker to monitor MODEM reception and normal operation. J2 allows monitoring of both the received and transmitted MODEM signals which may be fed to a tape recorder etc. The tape may be played back through J1 to test the system. J4 connects to the transceiver key jack through a shielded cable.

THE MODEM CIRCUITRY BREAKS DOWN INTO SECTIONS AS FOLLOWS:-

1. U1-(XR2206-IC) and associated circuits is the modulator.
2. U2-(XR2211 IC) and associated circuits is the demodulator.
3. U4-(LM324 IC) is a 2210-hz FILTER/500-hz BANDWIDTH.
4. Q5/Q6 and circuit comprise the CW receive interface.
5. U3-(4N28) OPTO isolator and Q1/Q2 switch the receive and transmit signals and provide isolation.

The demodulator - the XR2211 is a phased lock loop (PLL) decoder. Adjust the VCO tune control (PIN-12) to 2210-hz with the DMF on PIN-11, or J1 to J2 and adjust to the half way point between the red led just lighting and just going out. In operation the red led (PIN-5) is lit by means of Q3 and Q4 when a mark is received, and the audio input to J1 should be kept as low as possible. Correct tuning of the receiver is indicated when the red led is at it's brightest.

The FILTER - this is a simple 3-stage filter leaving one section of the LM324 unused. This could be wired for a further 2210-hz section, or dispensed with, in which case input would be to PIN-2 of the XR2211. However for best results a seperate mark filter centred on 2295-hz for 170-hz shift should be used.

CW INTERFACE- this is a simple circuit to rectify and amplify the CW input, the 2N3906 amplifier also being used to boost RTTY input to the computer. The green led will indicate correct CW tuning when at it's brightest. Of course the green led will also glow with RTTY signals, and the meter (not absolutely necessary), gives a further indication of correct tuning. There is no significance in the colours chosen for the led's. Two different colours seeming a better choice than two of the same.

OPERATION - OUTPUT FROM THE COMPUTER - signals from PIN-4 RS-232 socket, are fed to PIN-4, 5-PIN DIN socket to S6. If S6 is set to TTY, the TTY printer is activated to produce a hard-copy printout. With S6 set to trans and S3 at CW or RTTY/TRANS current flow lights the led in the OPTO coupler thus turning on the photo transistor. If S3 is set to CW, this keys the transceiver, or if set to RTTY/TRANS puts a short on PIN-9 of the XR2206 producing a mark tone of 2975-hz, set by S2. If S3 is set to CW, then when current flow ceases the transceiver is no longer keyed. The pattern of characters sent by the computer are translated into terms of dots and dashes for CW, or marks and spaces for RTTY, and in the latter case are outputed from PIN-2 of the XR2206 and fed to the microphone input to the transceiver via S3B, at 100% duty cycle since S3A (RTTY/TRANS), shorts out the transceiver TRANS/REC relay.

OPERATION - INPUT TO THE COMPUTER - received signals are fed via J1 to the filter, where hopefully only those signals within the 500-hz bandwidth centered on 2210-hz will pass. If S3 is set to CW, Q5 circuit rectifies and Q6 amplifiers the signal. The green lead is triggered which together with the meter, gives an indication of correct tuning. From here the signal is fed to PIN-5 5-PIN DIN socket and PIN-2 RS-232 computer socket, to be resolved by the program and be printed out on the monitor or TV screen in plain language. If S3 is set to SSB/RTTY rec then, after passing through the filter, signals are fed to the XR2211. When a mark is received the red led is lit and Q1 turns on. Q2 is used for reverse operation (if the demod. has latched onto a space or if the transmitting station is using reverse operation. Q2 is also switched into circuit by S3D during transmitting). Since received signals also trigger the OPTO coupler and XR2206 mod. a clean copy of the AFSK input to J1 may be recovered at J2. Output from the collector of J1 may be used to drive a TTY printer at J3, or fed via S3G and S3F to the 2N3906 amplifer. As with CW signals the green led will glow and the meter will give an indication of tuning, and the RTTY character will be processed by the program and a printout on the monitor or TV screen in plain language will be the result.

As previously mentioned both the received information, and that for transmission are stored in receive and transmit buffers within the program, and these may be DUMPED (to printer, tape or disc) when full, or as required.

LIST OF COMPONENTS

CAPACITORS:-

C1,C5,C9,C15 - 10UF TANTALUM C2 - 0.047UF MYLAR
C3 - 1UF 25V TANTALUM
C6,C7,C16 - 0.01UF DISC
C4,C8,C12,C17,C18 - 0.1UF DISC CERAMIC
C10,C11, - 0.0033UF DISC
C13 - 0.05UF 50 DISC CERAMIC
C14 - 0.022UF MYLAR

DIODES

D1,D2 - LED'S
D3 - IN914 OR SIMILAR
D4,D5,D6,D7,D9 - IN4003
D8 - IN914

INTEGRATED CIRCUITS

U1 - XR2206 FUNCTION GENERATOR
U2 - XR2211 DECODER
U3 - 4N28 OPTO COUPLER
U4 - LM324 QUAD OP AMP

PLUGS AND SOCKETS

J1,J1A,J2,J3,J4,J5 - MINIATURE JACKS (SOCKETS) WITH PLUGS
J6 - 4-PIN DIN SOCKET
J7 - TRANSCEIVER MIC SOCKET
J8 - 5-PIN DIN SOCKET
J9 - COMPUTER RS-232 SOCKET
P1 - 4-PIN MIC PLUG
P2 - 4-PIN DIN PLUG
P3 - 5-PIN DIN PLUG

RESISTORS (NOTE: ALL RESISTORS 5% 1/4 WATT)

R1,R2 - 5.1K OHM
R3,R35 - 50K OHM 1-TURN TRIMPOT
R4,R8 - 7.5K OHM
R5,R7,R9 - 5K OHM 10-TURN TRIMPOT
R6 - 4.7K OHM
R10 - 220 OHM
R11 - 1M OHM 1-TURN TRIMPOT
R12 - 18K OHM
R13,R18,R26 - 100K OHM
R14 - 82K OHM
R15,R27 - 220K OHM
R16,R19,R22,R34,R41 - 1K OHM
R17,R21,R30,R31,R36 - 10K OHM
R20,R23,R39 - 470 OHM
R24,R37 - 3.3K OHM
R25 - 470K OHM
R28 - 18K OHM
R29 - 10K 10 TURN TRIMPOT
R32 - 5.6K OHM
R33 - 330 OHM
R38 - 6.5K OHM
R40 - 33K OHM
R42,R43 - 3.9K OHM

SWITCHES

S1 - SINGLE THROW 12V DC POWER
S2 - 2-POS. 1-BANK
S3 - 3-POS. 7-BANK
S4 - 3-POS. 2-BANK
S5 - SINGLE THROW
S6 - 2-POS. 2-BANK

TRANSISTORS

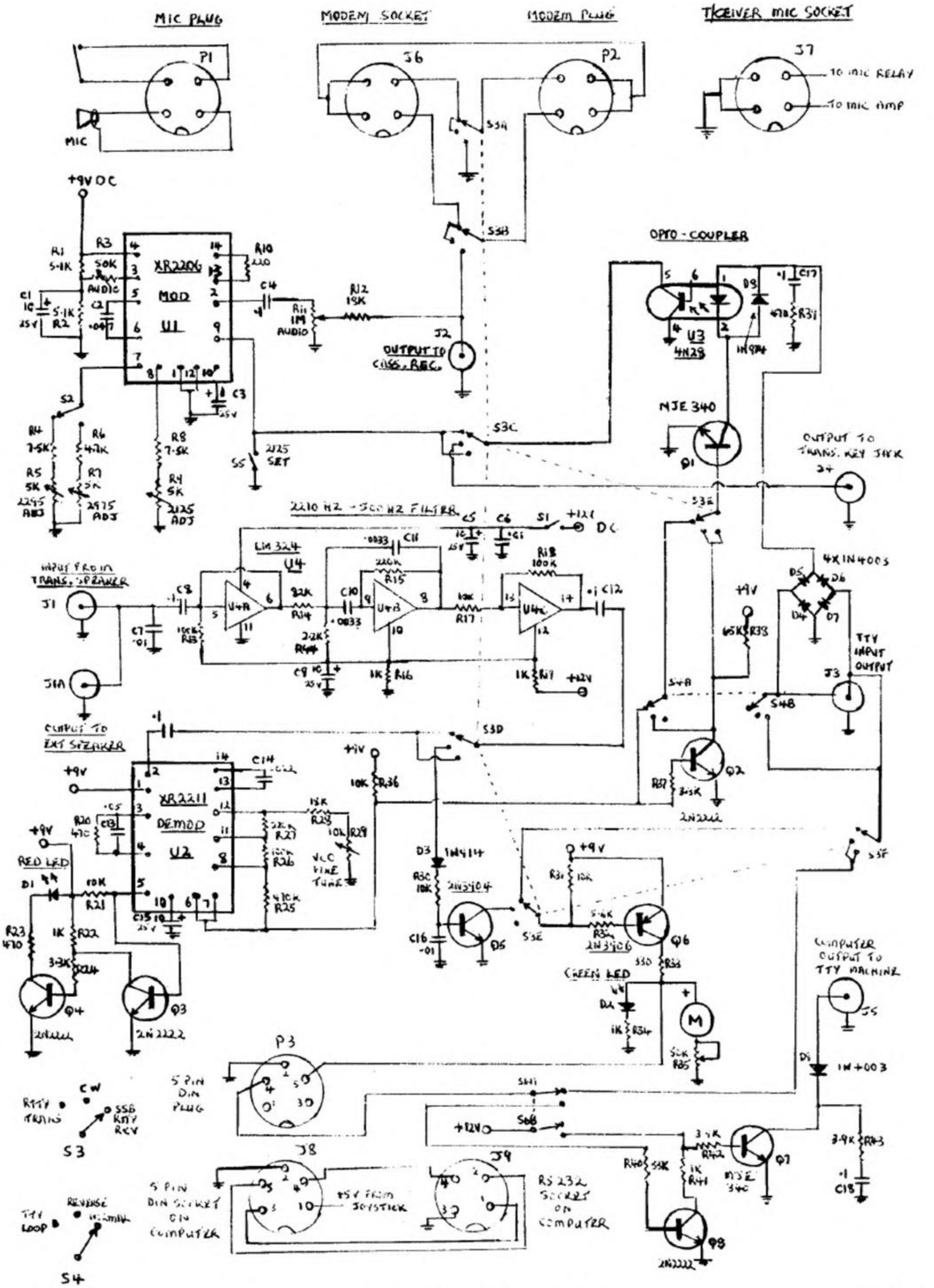
Q1,Q7 - MJE 340
Q2,Q3,Q4,Q8 - 2N2222
Q5 - 2N3904
Q6 - 2N3906

The MODEM may be constructed on a piece of veroboard or similar prepared PIN-BOARD. For ease of access for adjustment, all TRIMPOTS should be mounted along one edge of the board. My method is to plan the placement of parts on a piece of graph paper before construction. This is done on a larger sheet than the PIN-BOARD, but to scale. My front panel has the meter slightly higher than centre, with the two led's above the meter. S4 is on the left and S3 on the right of the meter, with S1,J6 and S2 in line below the meter. S5, a slide switch, is below J6 which is centre below the meter.

My rear panel has an input power socket and 6-miniature jacks mounted at the bottom. I use shielded cables fixed at the MODEM end, for P2 and P3 and these come out of the rear panel on either side thr sockets. S6 is mounted top right hand side of the rear panel. I folded a piece of aluminium 46-cm by 18-cm to a 'U' shape with 15-cm by 14-cm base, and 14-cm by 14-cm front and rear panels, all with 1.3-cm fold-in sides and a 'U' shape top cover to suit, attached with self tappingscrews. The PIN-BOARD, veroboard etc. Should be cut slightly less on each side, than the 15-cm by 14-cm allowed by the aluminium base. Stand-offs should be used to insulate the board from the chasis, and 4-rubber feet fitted to the base.

Finally, best of luck to those who

continued on P43..


MIC PLUG
MODEM SOCKET
MODEM PLUG
T/CEIVER MIC SOCKET
P1
J6
P2
J7
MIC
TO MIC RELAY
TO MIC AMP
+9V DC
XR2206
MOD
U1
OUTPUT TO CASS. REC.
OPTO-COUPLER
U3
4N28
NJE 340
2210 HZ - 500 HZ FILTER
LM 324
U4
INPUT FROM TRANS. SPEAKER
OUTPUT TO EXT SPEAKER
OUTPUT TO TRANS. KEY JACK
4X IN4003
TTY INPUT OUTPUT
XR2211
DEMOD
U2
RED LED
GREEN LED
VCO FINE TUNE
COMPUTER OUTPUT TO TTY MACHINE
5 PIN DIN PLUG
5 PIN DIN SOCKET ON COMPUTER
+5V FROM JOYSTICK
RS 232 SOCKET ON COMPUTER
S3
S4
CW
RTTY TRANS
RTTY RCV
TTY LOOP
REVERSE
NORMAL

CoCo LINK

WHY do I need a MODEM???

This is a question many of you have asked either at User-Group meetings or of other CoCo owners. The fact that you have a personal computer proves you have the intelligence and the inquiring mind required to grasp the nettle of the increasing technology of today's world.

Up until now you, the CoCo owners of Australia, have laboured, largely alone, trying to learn from and be mentally stimulated by the power of Tandy's little gem. The diverse fields you have explored, as evidenced by the range of articles and programs in each Australian CoCo, continues to grow.

The greatest stimuli for most is reading magazines and talking to others at User-Group meetings. However the magazine is largely one-way communication and the group meetings have time and distance limitations.

Now comes the next big step forward! With a modem connected to your computer, you and it can talk not only to other users at a time and pace suitable to you, but you can access bulletin boards around the country run by various organisations or best of all your own CoCo LINK with software ready to run on your system.

Those of you who read the American magazines will know of BBS such as Compuserve which have a multitude of 'special interest groups' for education, music, stamp collecting, graphics, stock market, sports etc. Hopefully in time, many of these will be available on COCO LINK.

Another interesting development referred to in last month's issue was the announcement by the Commonwealth Bank of the opening of 'BANKLINK' service in February. This is the first of many organisations offering data services to anybody with communication facilities. To use their service you will need a modem / keyboard / T.V. unit they will sell or better still use the existing hardware of your CoCo and its modem.

The answer to the question 'WHY'? is clear, the next is 'WHEN'? The sooner the better! COCO LINK will be a great asset to CoCo owners but to offer service it has to have input from users. The faster the numbers grow the better it becomes.

The third question is 'WHICH' modem to buy? There are a number of units available that will connect to CoCo and several software packages on sale, (with varying reliability, as covered in the October article).

The best solution is probably the CoCo MODEM available from RAINBOW BITS of 17 Penley St., THE GAP. Q. 4061. (I have no financial interest in RAINBOW BITS, but am happy for Brian and his happy band of workers to be using a related name! G.)

The CoCo Modem was designed locally to overcome the problems of the limited control facilities of Tandy's 4 wire printer I/O port. It has a ready to run terminal package included on an EPROM (no loading, it's ready to run on power-up) and also because it leaves the printer port free (no messy RS232 switch box to buy) you can continuously dump incoming listings etc. straight to your printer to read at your leisure.

It is a Telecom permitted attachment and will run on ALL versions of CoCo without extra power-pack units. CoCo MODEM is available in two versions, the STANDARD at $250 and the AUTOMATIC at $295. Both allow set up of all communication parameters, such as Number of bits, Stop bits, Parity etc. from a screen menu by single key commands. The AUTOMATIC has the added features of being able to auto-dial or answer incoming calls. (You could even set up your own BBS.)

It is simple to install, plug it into your cartridge port (or separately available Y-cable or multi-pak connector if you have disks) plug the cord supplied into the phone socket and you are ready to go.

This is one of the most interesting additions to your CoCo that you could contemplate. The added flow of ideas and

information thru the network will greatly add to your use and enjoyment of your computer.

The use of STD time may deter you slightly but the cost of this in the economy night periods is constantly falling and with the facilities available on this unit to dump data to printer or tape/disk for later perusal, on-line time can be kept to a minimum.

****

CoCo Link is taking shape - and is to schedule even!

Unfortunately, I was a bit slow getting the application to Telecom for a line, so I am unable to inform you of the number to call, however a quick call by phone here first, on our regular number will fix that problem.

We are not satisfied with the speed of the BBS package, but we think that we will be able to speed that given a little time and experience.

Remember, CoCo Link is free till 15th January, 1985. So there is no need to have a number or password at this stage. After 15th January, you will need to be a subscriber to get much from CoCo Link.

RTTY

continued from P41..

tackle the construction, and a further point for those who wish to write their own RTTY program. On page 62 of the Feburary 1983 issue of the AUST. RAINBOW, Dan Downard in his excellent article on this subject, gives a formula to determine BAUD constant. (55930/BAUD rate)-5 = BAUD constant. The result for 45.45-BAUD would be:- (55930/45.45)-5 = 1226 DEC. = 04CA HEX. so that POKE 149,4:POKE 150,& HCA gives a good starting point for experiment in obtaining a 45.45 BAUD output from the CoCo RS-232 PORT. In this respect a study of page 209/210 of getting started with Extended Color Basic; 'THE FACTS' for the color computer), the Radio Shack Color Computer Technical Reference Manual and the 'ROM EXPOSED' will all help to achieve the impossible.

SCOREBOARD

ASTRO BLAST (Mark Data)
David Coleman Yeronga 52080

ASTRO LANDER (CoCo Software)
R Boxall 4250

ATOM (Tandy)
David Thurbon (round.1) xe

BEAM RIDER (Spectral)
David Thurbon Canberra 83530

CALIXTO (Mark Data)
J Gans Bris 162

CANYON CLIMBER (Tandy)
Steve Lemke Bribie7/101080

DEVIOUS (Spectral)
R Boxall 28820

DONKEY KING (Tom Mix)
R Boxall 59300

FIRECOPTER (Adventure Intl.)
R Boxall 69152

FLYBY (Chromasette)
David Coleman Yeronga 32080

GALACTIC ATTACK (Tandy)
Ian Choat Woodridge35870

GALAX ATTACK (Spectral)
David Coleman Yeronga 27950

GHOST GOBBLER (Spectral)
Steven Marks Yanco L8/68250
Chris Nagle Leeton L18/58868

KATAPILLAR ATTACK (Tom Mix)
Steven Marks Yanco 9412

LANCER (Spectral)
M. Bloomfield Sydney 148650

LASERWORM (Rainbow)
Glynn Catherall Gold Co 30366

LUNAR ROVER PATROL (Spectral)
C Boxall 64480

MEGABUG (Tandy)
Lori Lehane Penrith 19540

MICROBES (Tandy)
Steven Marks Yanco L3/35410
Jack Rae Mt Isa 1/10700
R Boxall 63100
R Boxall & D Kemp 59600

MONSTER MAZE (Tandy)
Neil Prince Forbes 8410

MOON SHUTTLE (Data Soft)
David Thurbon Canberra 27700

PLANET INVASION (Spectral)
David Coleman Yeronga 48500

POLARIS (Tandy)
Neil Prince Forbes 13840
Chris Nagle Leeton 27563

POLTERGEIST (Tandy)
Steven Marks Yanco 4455

POOYAN (Datasoft)
R Boxall 100850

POPCORN (Tandy)
Chris Nagle Leeton 1/58120
Allan Rae MtIsa 56770

PYRAMID (Tandy)
J Gans Bris 200

RAAKATU (Tandy)
J Gans Bris 40

ROBOT BATTLE (Spectral)
R Boxall 10/4850

SEA QUEST (Mark Data)
J Dougan &J GansBris 165

SHENANIGANS (Mark Data)
J Gans Bris 112

SHOOTING GALLERY (Tandy)
Chris Lemke Bribiels 22420

SKIING (Tandy)
Jack Rae MtIsa 0:36.00

SPACE ASSULT (Tandy)
Neil Prince Forbes L1/4980

SPACE SHUTTLE (Tom Mix)
C Boxall 192

TIME BANDIT (Michtron)
J Dougan Bris 35860

TRAPFALL (Spectral)
David Thurbon Canberra 47910

TUT (Ardvark)
Keith Savage 99430

WHIRLYBIRD RUN (Spectral)
R Boxall 42375

WILDCATTING (Image Producers)
R Boxall 34692

ZAXXON (Tandy)
Jason Cook StClair 104600

RAINBOW WARE

This range of clothing is made here on the Gold Coast.

DESCRIPTION	SIZES	PRICE
V-Neck Crew Neck	12. 14. S. M. L. XL	14.95
Tank Tops	10. 12. 14. S. M. L. XL	13.85
Tennis Dress	S. M. L.	26.20
Hooded Zip-up Jacket	S. M. L.	31.95
Ladies Hooded Pullover Top	S. M. L.	28.50
Ladies Shorts	S. M. L.	17.95
Children's Tank Tops	6. 8. 10.-12. 14	12.80
Children's T-Shirts	8. 10. 12. 14	14.20
COLOURS AVAILABLE	white, pinik, aqua, sky blue, yellow, hot pink, black	

PLEASE SEND ORDER FORM

_____ $ _____

_____ $ _____

_____ $ _____

_____ $ _____ AMOUNT _____

Mr ☐ Mrs ☐ Miss ☐ Initials ☐☐ Surname

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

Subscription No.

Address

PC

☐ CASH
☐ CHEQUE
☐ POSTAL ORDER

☐ VISA BANKCARD ☐

_____ Cardholder

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

Now is the time to subscribe to Australian CoCo/MiCo

Copies of back issues can be obtained, subject to the availability of stocks, by using this order form and marking clearly which issues you require to be sent to you. Each issue costs $3.45 including postage and packing. Please enclose your cheque/postal order made payable to: Graham Morphett, PO Box 1742, Southport, 4215.

Send off the slip below to ensure that you get on the mailing list for JANUARY

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

Subscription Rates

AUSTRALIAN CoCo/MiCo			AUSTRALIAN RAINBOW/GoCo
☐	Latest	3.45 per copy	☐
☐	6 months	$19	☐
☐	12 months	$31	☐

Back Issues

$3.25 per copy

☐ 1st Issue '83 ☐ 1st Issue '81

DISKS & TAPES

Rainbow on Tape (program listings) $12 for month of ☐

Please note that RAINBOW on TAPE is issued irregularly

Tape Monthly CoCoOz			MiCoOz		
☐	Latest	$6	☐	Blank tapes 12 for $18 or $1.70 ea.	☐
☐	6 months	$32	☐	Cassette Cases	10 for $5 ☐
☐	12 months	$58	☐	Disks — $3.50 ea. ☐	10 for $28.99 ☐

BOOKS

Byte Elementary	$5.95	☐
Help Medium	$9.95	☐
Facts Advanced	$11.95	☐
MiCoHelp Medium	$9.95	☐
MiCo Exposed V. Advanced	$11.50	☐

BULLETIN BOARD

CoCo Link — Annual Sub. $29 ☐

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

☐ VISA BANKCARD ☐

_____ Cardholder

☐ CASH
☐ CHEQUE
☐ POSTAL ORDER

New Subscription ☐ Renewal ☐

If you already subscribe to either Australian Rainbow or Australian CoCo please place Subscription No.

BLOCK CAPITALS PLEASE

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

Mr ☐ Mrs ☐ Miss ☐ Initials Surname

Address

PC

Telephone Number STD Code Local Number

CRYSTAL BLADE Software

QUALITY SOFTWARE FOR THE TRS-80 COLOR COMPUTER

P.O. BOX 256, ROSEVILLE. NSW. 2069

LARGEST RANGE OF SOFTWARE IN AUSTRALIA - PROMPT MAIL ORDER SERVICE

SEND $2 FOR COMPLETE CATALOGUE (OVER 30 PAGES)

ROSEVILLE USER'S GROUP MEETING. ENQUIRIES PHONE: (02) 467-1619

..... PROGRAMMING PROBLEMS - I/Output Errors on loading; Software Operation queries - there is sure to be someone there with years of experience with the COCO who would be happy to assist you and give advice.

Arcade 16K

ASTRO BLAST	31.00	LAS VEGAS	22.00
ASTRA LANDER	22.00	LEMANS	32.00
BEAM RIDER	31.00	LIGHT RUNNER	31.00
BLOCHEAD	31.00	MAD BOMBER	22.00
BUMPERS	31.00	MOON SHUTTLE	36.00
BUTTERFLY BOMBER	32.00	MS NIBBLER	31.00
CATERPILLAR	31.00	TRAVELIN TOAD	31.00
CAVE HUNTER	31.00	NOVA PINBALL	25.00
CESSNA HUNTER	22.00	PENGON	32.00
CHOPPER STRIKE	34.00	PLANET INVASION	31.00
COLOR FURY	34.00	RAIL RUNNER	31.00
COLORPEDE	37.00	ROBOT BATTLE	31.00
COSMIC CLONES	31.00	SEAWOLFE	24.00
DEFENSE	27.00	SHARK TREASURE	31.00
DEMON ASSAULT	31.00	SPACE INVADERS	31.00
DOODLE BUG	31.00	SPACE RACE	31.00
EL BANDITO	31.00	SPACE RAIDERS	31.00
FLIPPER	22.00	SPACE SENTRY	27.00
FLYING TIGERS	31.00	SPACE TRADER	22.00
FOOTBALL	31.00	STRIP TEASE	18.00
GALACTIC HANGMAN	20.00	SUB HUNT	24.00
GHOST GOBBLER	31.00	TRAPFALL	34.00
GLAXIONS	31.00	WHIRLYBIRD RUN	31.00
HAYWIRE	31.00	XYGOID	31.00
INTERGALACTIC FORCE	27.00	YAHTCC	16.00
KATERPILLAR ATTACK	31.00	ZEUS	24.00
KOMET KAZE	31.00	ZONE SIX	31.00

Arcade 32K

CASHMAN	34.00
CHOPPER STRIKE	34.00
COLOR FURY	34.00
CUBIX	31.00
DEMON SEED	34.00
DESERT PATROL	31.00
DEVIOUS	31.00
DONKEY KING	32.00
EIGHT BALL	31.00
FLIGHT	27.00
FOOD WAR	31.00
ICEMASTER	31.00
INTERCEPT 4	34.00
LANCER	31.00
LUNAR ROVER PATROL	31.00
MS GOBBLER	31.00
MUDPIES	34.00
OUTHOUSE	34.00
POOYAN	36.00
ROBOT BATTLE	31.00
TIME BANDIT	34.00
TUT	31.00
WACKY FOOD	31.00
ZACKSUND	32.00
ZAXXON	43.00

Adventures

ADVENTURES IN WONDERLAND	31.00
BLACK SANCTUM	22.00
CALIXTO ISLAND	22.00
ERLAND	31.00
GANGBUSTER	31.00
GREYMOON	22.00
INCA TREASURE	22.00
KEYS OF WIZARD	25.00
LOTHAR'S LABRYNTH	16.00
MONSTERS & MAGIC	31.00
PARANOIDS ANONYMOUS	22.00
PIRATES AHOY	16.00
PYRAMID	27.00
QUEST	27.00
SQUIRE	16.00
VIKING	31.00

Graphic Adv.

ACROSS THE RUBICON	22.00
ADVENTURE TRILOGY	27.00
BEYOND CIMEEON MOON	27.00
CALIXTO ISLAND	31.00
CONQUEST OF KZIRGLA	27.00
DUNGEONS OF DEATH	27.00
FEMBOTS REVENGE	27.00
INSPECTOR CLUESEAU	30.00
LABRINTH	27.00
SCEPTRE OF KZIRGLA	22.00
SEAQUEST	31.00
SHENANIGANS	31.00
TREK 16	25.00
WIZARD 64	27.00
WIZARDS TOWER	27.00

Future

SUPER ASTROLOGY	31.00
I CHING	22.00
NUMEROLOGY	25.00
TARROT	27.00

Utilities

64 COL MOD I/III EMULATOR	20.00
AUTO RUN	21.00
BUGOUT MONITOR	27.00
COCO CALLIGRAPHER	31.00
DISASSEMBLER	27.00
DISK MANAGER	36.00
DISK UTIL	25.00
GRAPHIC ANIMATOR	16.00
MAGIC BOX	28.00
PLUS 32	20.00
PRITTY PRINTER	25.00
QUICK SORT	16.00
SCREEN MACHINE	37.00
SOUND SOURCE	27.00
SUPER SCREEN	37.00
SUPER ZAP	42.00
TAPE UTIL	42.00
THE COMPOSER	31.00
VIP DISK ZAP	70.00
VDOS / VDUMP	34.00

Applications

AUTOTERM	48.00
CC DATABASE / LETTERWRITER	54.00
CC WRITER DISK	42.00
CC WRITER TAPE	25.00
COLOR DFT	32.00
DISK ZAPPER	48.00
ELITE CALC	65.00
ELITE WORD	70.00
FILMASTR	42.00
GRAPHICS EDITOR	22.00
MAILING LIST	59.00
MAIL / MERGE	31.00
STATGRAF	25.00
SUPER SCROLL	31.00
TELEWRITER-64	59.00
TERMTALK	47.00
VIP CALC	65.00
VIP DATABASE	70.00
VIP SPELLER	70.00
VIP TERMINAL	54.00
VIP WRITER	65.00
TIMS	31.00

RAINBOW BITS

supplier of the latest hardware add-ons for CoCo

CoCo MODEM

AUTOMATIC $295 STANDARD $250

1. Telecom approved to 300 baud bi-directional modem
2. Plugs into Rom port ex expansion pack to work with disk
   (Specify Tandy controller or other)
3. Inbuilt terminal programme on Eprom (no loading)
4. Automatic model allows CoCo to dial-up remote computer/B.B.S
5. Requires no power pack. Works with all models of CoCo.
6. Leaves printer port free for continuous down load to printer

Y CABLE $30
Permits simultaneous connection of modem and disk.

4-SLOT EXPANSION PACK Kit $65 (requires soldering) Made Up $100
Enables disk controller, modem, 80 column card, eprom programmer, rom packs etc, to be run/selected at one time

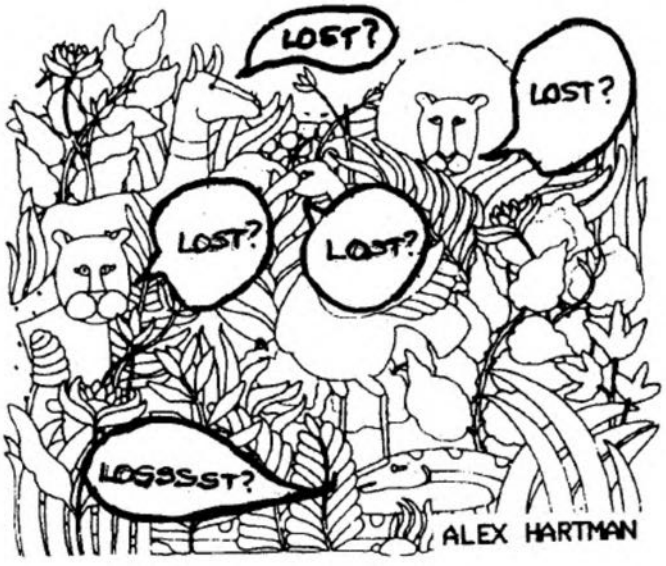
CASE CHANGER
LOWER CASE BOARD made and tested $60 (simple installation)
gives true lower case (with descenders) on screen not reverse video

VIDEO AMP CARD $25 (simple installation) for driving video monitor

all prices include P & P.

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

RAINBOW BITS



LOST IN THE DESERT is a 16K EXTENDED COLOR BASIC adventure. It is an ancient program from my "dark era". In this program the instructions are simple and could be the basis of the average T.V. melodrama.

Your plane crashes in the desert and you have 1 litre of water. The pilot has notified search and rescue. Within the desert there are an oasis, poisoned oasis, cannibals, poisonous snakes and cliffs. Oh! there are also 2 search parties wandering about.

Each move takes up .1 of your water supply, snakes take .5 while cliffs take .2. Finding a non-poisoned oasis will get you .5 extra.

The commands are simple; N=north, S=south, E=east, W=west and 2 special functions D=die (when you're ready to give up) and AHH! for those extreme emergencies gets you .5 litres of water extra. You have 2 AHH's.

Cannibals! Walk into thein camp twice and you'll be invited to dinner (heh-heh)!!

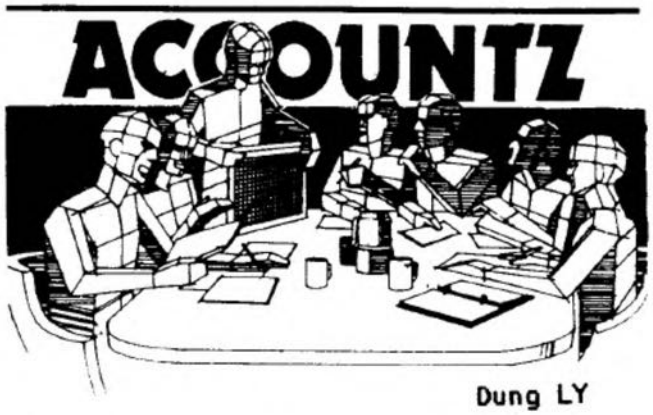
One special feature of this program is the title. After LOADing but before RUNning turn the volume up about half way and press play. By the way it's called CARAVAN. You'll know it when you hear it.

The Listing:

```
  5 'LOST IN THE DESERT
  10 'MADE IN MAY, 1983
  15 'BY 'THE PROS, INC'
  20 CLEAR1000:CLS:PRINT@7,"LOST I
N THE DESERT":PRINT@34,"PLEASE W
AIT - PROCESSING MAP!":DIMA(20,2
0):FORI=1TO20:FORJ=1TO20:READA(I
,J):NEXTJ,I
  25 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,5,0,6,0,0,0,0,5
  30 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0
,3,3,3,3
  35 DATA0,0,0,0,0,2,0,0,0,0,2,0,0
,0,0,3,5,0,0,3,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,3,0,0,1,3,0,0,3,3,3
,3,0,0,5,0,0,0,0,0,0,0,0,0,0,0,0
,0,3,0,0,0,3,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0
,0,0,1,0,0,2,0,0,0,2,0,5,0,0,0,0
,0,5,0,1,0,0,0,0,0,0,0,0
  40 DATA0,0,0,0,0,0,0,3,3,0,0,0,0
,0,0,0,0,0,0,5,0,0,0,0,0,0,0,3,5
,3,3,3,4,0,0,3,5,0,3,0,0,0,7,0,0
,0,0,0,0,0,0,3,0,5,0,3,3,3,3,0,0
,0,0,0,0,1,5,0,0,0,5,3,0,0,0,0,0
,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,2,0,0,0,0,0,0,0,5,0,0,5,0
,0,0,0,0,0,0,0,0,0,0,5,0,0
  45 DATA0,0,3,0,0,0,0,0,2,0,0,0,0
,6,0,0,0,0,0,0,0,1,3,0,3,0,0,0,0
,0,5,0,0,0,0,0,0,0,0,0,3,3,3,0,3
,0,5,0,3,0,0,0,0,0,1,0,0,5,0,0,0
,0,0,0,0,3,3,3,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,5,0,0,0,0,4,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,9
  50 CLS0:A3$=STRING$(192,255):A1$
=STRING$(96,223):A2$=STRING$(160
,159):PRINT@0,STRING$(176,223):P
RINT@176,STRING$(176,223);:PRINT
@352,STRING$(128,159);:PRINT@135
,"LOST IN THE DESERT";:Y=11:X=10
:WAT=101:C$="DESERT"
  55 AUDIOON:MOTORON:FORI=1TO10800
:NEXT:MOTOROFF
  60 CLS:PRINT@7,"LOST IN THE DESE
RT":PRINT:PRINT"INSTRUCTIONS";:I
NPUTI$:IFLEFT$(I$,1)="N"THEN95
  65 CLS:PRINT@7,"LOST IN THE DESE
RT":PRINT:PRINT"YOUR PLANE HAS C
RASHED IN THE    DESERT AND YOU H
AVE ONE LITER    OF WATER. YOUR A
IM IS TO GET     OUT OF THE DESER
T BEFORE YOU     DIE OF THIRST."
  70 PRINT"THERE ARE TWO SEARCH PA
RTIES    AND TWO TRIBES OF CANNI
BALS.
  75 PRINT"IF THE CANNIBALS GET YO
U THE    SECOND TIME, YOU DIE. T
HERE ARE ALSO SNAKES. THESE ARE
         POISONOUS AND WILL TAKE
 AWAY .5 LITER FROM YOUR WATER S
UPPLY.":PRINT"PRESS ENTER"::INPU
```

```
TI$:CLS:PRINT@7,"LOST IN THE DES
ERT"
80 PRINT:PRINT"THERE ARE ALSO OA
SIS HERE, BUT"
85 PRINT"THE CANNIBALS POISONED
SOME OF  THEM. WHEN YOU REACH A
NON-     POISONED OASIS, YOU GET
 HALF A  LITER WATER.":PRINT"THE
RE ARE CLIFFS HERE AND WHEN  YOU
 ENCOUNTER ONE, YOU LOSE .2  OF
YOUR WATER TO CLIMB IT."
90 PRINT:PRINT"PRESS ENTER";:INP
UTI$
95 CLS:PRINT@7,"LOST IN THE DESE
RT":PRINT:PRINT"YOUR COMMANDS: (
CMD)":PRINT"N: GO NORTH":PRINT"S
: GO SOUTH":PRINT"E: GO EAST":PR
INT"W: GO WEST":PRINT"D: GIVE UP
":PRINT"AAH: EXTRA 1L OF WATER (
2*ONLY!)":PRINT"PRESS ENTER TO S
TART";:INPUTI$
100 GOSUB245
105 WAT=WAT-1:IFWAT=0THENM$="DIE
D":GOTO300
110 IFWAT=0THEN300
115 PRINT@352,STRING$(32,32);:PR
INT@352,"YOU SEE: "C$;:PRINT@384
,"WATER:"WAT/10:LINEINPUT"CMD>";
CM$
120 IF CM$=""THENC$="EH?":GOTO11
5
125 IFCM$="W"THEN160
130 IFCM$="E"THEN170
135 IFCM$="N"THEN180
140 IFCM$="S"THEN190
145 IFCM$="D"THEN200
150 IF CM$="AAH"THEN AAH=AAH+1:I
F AAH=3THENC$="SORRY, NO GO!":GO
TO115ELSE WAT=WAT+10:IF AAH=1THE
NC$="ONE 'AAH' LEFT!":GOTO115ELS
EC$="YOUR LAST 'AAH'...":GOTO115
155 C$="HUH? COME AGAIN?":GOTO11
5
160 Y=Y-1:IFY<0THENY=20
165 GOSUB205:GOTO105
170 Y=Y+1:IFY>20THENY=0
175 GOSUB205:GOTO105
180 X=X-1:IFX<0THENX=20
185 GOSUB205:GOTO105
190 X=X+1:IFX>20THENX=0
195 GOSUB205:GOTO105
200 M$="GAVE UP":GOTO300
205 IF A(X,Y)=0THENC$="DESERT":G
OSUB245:RETURN
210 ON A(X,Y)GOTO215,220,225,230
,235,240
215 C$="AN OASIS":SIT=1:GOSUB250
:RETURN
220 C$="A POISONED OASIS":SIT=2:
GOSUB260:RETURN
225 C$="A CLIFF (-.2 WATER!)":GO
SUB265:RETURN
230 C$="CANNIBALS LEFTOVERS":GOS
UB270:RETURN
235 C$="A SNAKE (-.5 WATER!)":GO
SUB285:RETURN
240 C$="A SEARCH PARTY":GOSUB290
:SIT=6:RETURN
245 CLS0:PRINT@0,A1$;A1$;A2$;:RE
TURN
250 GOSUB245:FORI=116TO212STEP32
:PRINT@I," ";:NEXT:FORI=114TO118
:PRINT@I," ";:NEXT:PRINT@145," "
;:PRINT@151," ";:IFSIT<>2THENWAT
=WAT+50:SIT=1
255 FORI=231TO237:PRINT@I,CHR$(1
75);:NEXT:FORI=262TO270:PRINT@I,
CHR$(175);:NEXT:FORI=295TO301:PR
INT@I,CHR$(175);:NEXT:RETURN
260 GOSUB250:RETURN
265 GOSUB245:FORI=167TO187:PRINT
@I,CHR$(255);:NEXT:FORI=137TO153
:PRINT@I,CHR$(255);:NEXT:WAT=WAT
-2:IFWAT<=0THEN300ELSERETURN
270 GOSUB245:Z=Z+1:IFZ=2THENFORI
=241TO248:PRINT@I,CHR$(128);:PRI
NT@I+32,CHR$(128);:NEXT:FORI=306
TO311:PRINT@I,CHR$(128);:NEXT:PR
INT@214,CHR$(191);:PRINT@147,"(Y
OU)";:M$="GOT EATEN":PRINT@352,"
THE CANNIBALS GOT YOU!":GOTO280E
LSE275
275 GOSUB245:FORI=200TO296STEP32
:PRINT@I,CHR$(128);:NEXT:PRINT@1
67,CHR$(203);CHR$(207);CHR$(199)
;:PRINT@200,CHR$(195);:RETURN
280 FORI=1TO1000:NEXT:REA$="CANN
IBALS GOT YOU!":GOTO300
285 GOSUB245:PRINT@269,"  ";:PRIN
T@302,"   ";:PRINT@272,"  ";:PRINT
@239,"  ";:PRINT@207,"  ";:PRINT@1
75,CHR$(140);:PRINT@206,"    ";:W
AT=WAT-5:IFWAT<=0THENM$="DIED":G
OTO300ELSERETURN
290 GOSUB245:PRINT@352,C$:PRINT@
384,"YOU'RE SAVED!":A5$=CHR$(222
)+CHR$(212):PRINT@175,A5$;CHR$(2
23);A5$;:GOTO305
295 GOTO295
300 GOSUB245:FORI=111TO208STEP32
:PRINT@I,CHR$(207);:NEXT:PRINT@1
42,CHR$(207);:PRINT@144,CHR$(207
);:PRINT@352,"YOU "M$" IN THE DE
SERT!"
305 PRINT@416,"WANT TO PLAY AGAI
N(Y/N)";:INPUTI$:IFLEFT$(I$,1)="
N"THENENDELSEWAT=101:Z=0:C$="DES
ERT":X=10:Y=11:AAH=0
310 GOTO95
```

We received the following Accounts programs, and were impressed with both. One lists to six pages and the other to three. Quandry. Worry. How to do justice to two good programs AND get them to you ASAP. ANSWER. Put articles in magazine and programs on CoCoOz!



Dung LY

There have been many comments from my friends since my first version of CoCo ACCOUNTANT was published on CoCoOz a few months ago. They said that it was unfair that my program could only be used with a printer, (well I wrote it that way because.I own a printer!).

After RUNning my CoCo ACCOUNTANT a few times, I felt that there was something missing in my program. So I decided to do something about it! After five hours work, I came up with ACCOUNT2. This second version contains many extra features that were not available on my first version.

The program is fairly well self-explanatory. It's very similar to to the previous version, but the data is checked as it is typed in. For example, in the INPUT MENU, at the end of each entry, the computer will ask "Is this correct?", so if you make a mistake you can re-enter that entry by simply pressing "N". Otherwise, an entry will not be accepted unless you press "Y".

Another useful feature is a protection feature to save you from entering data over already existing data in memory, whether the entry data be sourced from the keyboard or tape. If you want to overwrite data, press SHIFT 3 together when prompted.

When (4) is selected from the Menu, the computer will ask for your selection of screen or printer. You can also select baud rate for the printer. Pressing "1" or "2" will start the printer.

The program requires that you input the date. Printing will not proceed until you do so.

I would be happy to assist anyone who wants to write his own business programs.

by Bob La Maitre.

32K ECB, Printer

wklyacc

The program WEEKLY ACCOUNTS has been developed during the last 12 months and has gone through various stages of modification during this time.

It began with the setting up of a small retail fabric business for my wife last year. At the time our accountant gave her a double foolscap sheet of paper all neatly formalised with columns for our daily takings and expenditure, these all had to be totalised at the end of the week, as well as have a carbon copy of the same.

This was a messy and time consuming job which I thought "why not let CoCo handle this, it's more than capable."

So from there WEEKLY ACCOUNTS was born. After a while our accountant wanted to see the books so I gave him the printout from the computer. He was pleased to see an easy to read computer printout, but then he started to ask, "why not have this and that shown as well"? So WEEKLY ACCOUNTS was modified once again to include his requests.

Come tax time, half his work had been done for him, and when he presented his bill to us it was half of what he had estimated. This saving will hopefully give rise to the addition of a disk to my computer desk.

The program is menu driven in 11 basic steps and requires 32K ECB along with POKE25,6:NEW before loading. I have used the orange screen routine presented in an early edition of Australian Rainbow.

"WKLYACC"

The following is a sample printout of the main body of the printer routine.

```
                    OI'S KNIT FABRIC WORLD
                    THE HUB WYNYARD ST.
                    TUMUT ** 47-1538

  ACCOUNTS FOR WEEK ENDING  18/08/ 1984

TOTAL DEPOSITS TO DATE          $12,945.61
TOTAL CHEQUES TO DATE           $ 9,509.42

-------DEPARTMENT TOTALS TO DATE--------
PATTERNS          $    618.15
HABEDASHERY       $ 1,326.04
FABRICS           $ 6,008.71
THREAD/BUTTONS    $    689.36
MACHINES          $ 3,125.05
CLASSES           $    905.00
LAY-BY            $    621.86   $13,294.17
PETTY CASH        $    238.71
DISCOUNTS         $    109.85   $    348.56
                                $12,945.61

---------DAILY DEPOSITS TO DATE---------
MONDAY            $ 2,120.35
TUESDAY           $ 1,254.64
WEDNESDAY         $ 3,584.80
THURSDAY          $ 2,349.05
FRIDAY            $ 2,614.00
SATURDAY          $ 1,022.77    $12,945.61

UPDATED BANK BALANCE                    UPDATED CREDITORS BALANCE
OPEN BALANCE   $ 4,913.23               OPEN BALANCE   $ 3,442.83
ADD DEPOSITS   $ 1,870.08               ADD INVOICES   $   120.50
SUB TOTAL      $ 6,792.31               SUB TOTAL      $ 3,563.52
LESS CHEQUES   $   650.66               LESS CHEQUES   $   650.66
CLOSE BALANCE  $ 6,101.65               CLOSE BALANCE  $ 2,872.51

                WEEKLY CHEQUES

DATE      PAYEE          CHQ NO    AMOUNT      ACCOUNT
13/08/84  STATE BANK     456       $ 400.00    LOAN
15/08/84  TEXTILE IMP    457       $ 245.00    FABRICS
17/08/84  STATE BANK B/C ---       $   5.66    B/C CHARGES
   TOTAL WEEKLY CHEQUES             $ 650.66

            PETTY CASH TAKEN OUT THIS WEEK

DATE      AMOUNT      ACCOUNT
15/08/84  $   5.30    STATIONARY
17/08/84  $  10.00    FREIGHT
TOTAL PETTY CASH THIS WEEK  $  15.30

WEEKLY SALES 13/08/ 1984  TO 18/08/ 1984

PATTERNS          $   44.50     MONDAY    13/08 $  150.44
HABEDASHERY       $  185.34     TUESDAY   14/08 $  183.98
FABRICS           $  837.68     WEDNESDAY 15/08 $  851.87
THREAD/BUTTONS    $   78.74     THURSDAY  16/08 $  171.83
MACHINES          $  355.08     FRIDAY    17/08 $  323.21
CLASSES           $  350.00     SATURDAY  18/08 $  166.05
LAY-BY            $   98.54
SUB TOTAL         $ 1,888.38
- PETTY CASH      $   15.30
TOTAL SALES THIS WEEK HERE                      $ 1,873.08
```

previous weeks totals, this puts into memory all running totals for departments and daily deposits as well as the previous weeks closing bank balance and closing creditors balance.

Step 2 is where the daily department totals are entered. These totals are taken directly from the Z-total produced by the cash register at the end of each day and accumulated to the end of the week. First enter the date in the DD/MM format, and then the totals after each prompt. The discounts at the end are automatically taken care of but they only apply to the sale of fabrics, where we give a 10% discount to bona-fide Tech. College students.

Step3 is where the weekly expenditure is entered, there are several limitations to the way things can be entered. The date must be in the format DD/MM/YY eg. 05/12/84. The payee must have no more than 15 letters, so when the cursor is at the edge of the screen, please do not enter any more. The cheque No. uses only the last three digits of the cheque. The amount can be up to $99,999.99. The account can also have no more than 15 letters as per the payee.

When complete just enter ZZ in the date section, it will then display the total expenditure and go on to the petty cash routine. Just answer the prompts with regard to the same limitations as per the cheques section.

Step 4 enters the weeks invoices. It only requires a total figure of all the invoices, not all the details.

Steps 5 and 6 give a screen printout of daily and weekly details and totals.

Steps 7 and 8 produce a screen printout of the updated bank balances and updated creditors balance.

Step 9 saves to tape all relevant running totals ,for departments, daily deposits, total deposits and cheques, plus the closing bank balance and creditors balance. This must be carried out prior to printing the results as several vital calculations are carried out at this stage.

Step 10 saves all the cheque and petty

The first step is to load in the

cash expenditure in single strings which must be saved to a separate tape. This save is later used in my other program which gives me a complete breakdown of all expenditure by month, account and payee. This program will be presented in a later edition of CoCoOz.

Finally a complete breakdown of all totals and information is entered on the printer.

I use a CGP115 printer which I have found to be a reliable small printer which gives a neat printout of all the information required. It may be slow but at least I can have a cup of coffee while it does its work.

I hope this will be of some use to those of you who run a small business. at least you should be able to save a few 'bob' on your accountant fees.

16K ECB, Joysticks

by Steven Youngberry.

GUNFIGHT is a two player Shoot-Em-Up game of particular appeal to kids young and old.

Written to use the PMODE2 graphics screen the gameplay is simplicity in itself. The two Gunfighters face one another from either side of the screen. You are given six shots in each round. The player scoring the most hits on his opponent wins the round. The best of five rounds determines the overall winner.

Your only cover is a wall in the centre of the screen (the proverbial 'side of a barn') and in the true fashion of walls changes its size and location when least expected.

The Listing:

```
  10 CLS:PRINT:PRINT:PRINT:PRINT:P
  RINT"    WRITTEN BY"
20 PRINT"    S.YOUNGBERRY,"
30 PRINT"    OCTOBER 84."
40 FORT=1TO1000:NEXTT
50 CLEAR1000
60 PRINT:PRINT"    DO YOU WISH IN
STRUCTIONS."
70 B$=INKEY$:IFB$="N"THEN210
80 IFB$<>"Y"THEN70
90 CLS
100 PRINT@106,"GUN FIGHT"
110 PRINT" GUN FIGHT IS THE BEST
 OF FIVE    SHOOTS, A SHOOT IS TH
E FIRST    WITH SIX 'KILLS'."
120 PRINT:PRINT" NUMBER OF KILLS
 RESET AFTER      EACH SHOOT. GUN
S HAVE SIX       SHOTS AND ARE";
130 PRINT" RELOADED AFTER":PRINT
" EACH KILL OR WHEN BOTH GUNS AR
E EMPTY."
140 PRINT" THERE IS A SHORT TIME
 DELAY      AFTER EACH EACH 'KILL
' BEFORE    YOU CAN FIRE AGAIN"
149 PRINT:PRINTTAB(10)"<<ENTER>>
"
150 A$=INKEY$:IFA$=""THEN150
160 FORS=1TO16:PRINT
170 FORT=1TO100:NEXTT
180 NEXTS
190 POKE65495,0
200 DIMN$(10)
210 N$(1)="RNRU6NGBR2BD6"
220 N$(2)="NR4U3R4U3L4DBD5BR6"
230 N$(3)="NUR4U3NL3U3L4DBD5BR6"
240 N$(4)="BR3U6G3DR4BD2BR2"
250 N$(5)="BUDR4U3L4U3R4BR2BD6"
260 N$(6)="BU5BR4UL4D6R4U3L4BR6B
D3"
270 N$(7)="BU6R4DG4DBR2"
280 N$(8)="R4U3NL4U3L4D6BR6"
290 N$(9)="BUDR4U6L4D3R4BR2BD3"
300 N$(0)="R4U6L4D6BR6"
310 ZZ$="BU3NF2NE2R4BD3BR2"
320 ZX$="BU3R4NH2NG2BD3BR2"
330 ZH$="U6BD3R4NU3D3BR2"
340 ZT$="BR2U6NL2R2BR2BD6"
350 ZI$="RNRU6NLRBR2BD6"
360 ZK$="U6BD3RNE3F3BR2"
370 N$(10)="BL3RNRU6NGBD6BR3R4U6
L4D6BR2"
380 ZS$="BUDR4U3L4U3R4DBD5BR2"
390 ZL$="NU6R4BR2"
400 ZO$=N$(0)
410 ZR$="U6R3FDGL2F3BR2"
420 'GUNFIGHTER LOOP
430 PMODE2,1:PCLS:SCREEN1,1
440 DIMLM(10,20):DIMRM(10,20)
450 LM$="L2E2U2NL6NR2U2L6D2NL2D2
F2D12R5U2L2U5R3U2NR3L3U2"
460 RM$="R2H2U2NR6NL2U2R6D2NR2D2
G2D12L5U2R2U5L3U2NL3R3U2"
470 PCLS
```

```
480 LINE(1,20)-(254,190),PSET,B
490 DRAW"BM10,30"+LM$
500 DRAW"BM245,165"+RM$
510 GET(5,21)-(15,45),LM
520 GET(240,156)-(250,191),RM
530 RA=160
540 W=RND(20)+20
550 GOTO1000
560 'MOVE GUN FIGHTER
580 IFLS>6ANDRS>6THENLS=0:RS=0
590 Z=JOYSTK(0):X=JOYSTK(2)
600 A=JOYSTK(1):C=JOYSTK(3)
610 IFA>40THENB=+3 ELSEIFA<20THE
NB=-3 ELSEB=0
620 IFC>40THEND=+3ELSEIFC<20THEN
D=-3ELSED=0
630 LA=LA+B:RA=RA+D
640 IFLA<21THENLA=21ELSEIFLA>165
THENLA=165
650 IFRA<21THENRA=21ELSEIFRA>165
THENRA=165
660 PUT(5,LA)-(15,LA+24),LM
670 PUT(240,RA)-(250,RA+24),RM
680 SS=SS+1
690 IFSS<10THEN560
700 F=PEEK(65280)
710 W1=W1+1:IFW1=W THENW=RND(20)
+20:GOSUB1310
720 IFF=126ORF=254THEN750
730 IFF=125ORF=253THEN850
740 GOTO560
750 'LEFT GUN
760 LS=LS+1:IFLS>6THEN820
770 SOUND100,1
780 IFLA+12>W2 ANDLA+12<W3 THENL
INE(18,LA+12)-(124,LA+12),PSET:G
OTO840
790 LINE(18,LA+12)-(250,LA+12),P
SET
800 IFLA+12>RA ANDLA+12<RA+24THE
N830
810 LINE(18,LA+12)-(250,LA+12),P
RESET
820 GOTO560
830 PLAY"L17502CBA01GFED":LINE(1
8,LA+12)-(250,LA+12),PRESET:LH=L
H+1:GOTO950
840 LINE(18,LA+12)-(124,LA+12),P
RESET:GOTO560
850 'RIGHT GUN
860 RS=RS+1:IFRS>6THEN920
870 SOUND75,1
880 IFRA+12>W2 AND RA+12<W3 THEN
LINE(250,RA+12)-(128,RA+12),PSET
:GOTO940
890 LINE(239,RA+12)-(10,RA+12),P
SET
900 IFRA+12>LA ANDRA+12<LA+24THE
N930
910 LINE(239,RA+12)-(10,RA+12),P
RESET
920 GOTO560
930 PLAY"L17501GFEDCBA":LINE(239
,RA+12)-(10,RA+12),PRESET:RH=RH+
1:GOTO950
940 LINE(250,RA+12)-(128,RA+12),
PRESET:GOTO560
950 'RESULTS
960 LS=0:RS=0:SS=0
970 PCLS:LINE(1,20)-(254,190),PS
ET,B
980 IFLH>5THENFL=FL+1:GOSUB1060
990 IFRH>5THENFR=FR+1:GOSUB1060
1000 DRAW"S8BM10,17"+N$(LH)
1010 DRAW"S8BM230,17"+N$(RH)
1020 DRAW"S8BM65,17"+N$(FL)+ZZ$+
ZS$+ZH$+ZO$+ZO$+ZT$+ZS$+ZX$+N$(F
R)
1030 IFXX=1THENGOTO1190
1040 LINE(125,W2)-(127,W3),PSET,
B
1050 GOTO560
1060 QS=QS+1
1070 IFQS=1THENS1=LH:S2=RH
1080 IFQS=2THENS3=LH:S4=RH
1090 IFQS=3THENS5=LH:S6=RH
1100 IFQS=4THENS7=LH:S8=RH
1110 IFQS=5THENS9=LH:S0=RH
1120 LH=0:RH=0
1130 IFFL=3ORFR=3THEN1170
1160 RETURN
1170 PLAY"L10001FGAB02CP10CDEFP1
0FGAB"
1180 XX=1:GOTO1000
1190 DRAW"S8BM102,51"+ZK$+ZI$+ZL
$+ZL$+ZS$
1200 DRAW"S8BM109,77"+N$(S1)
1210 DRAW"S8BM141,77"+N$(S2)
1220 DRAW"S8BM109,103"+N$(S3)
1230 DRAW"S8BM141,103"+N$(S4)
1240 DRAW"S8BM109,129"+N$(S5)
1250 DRAW"S8BM141,129"+N$(S6)
1260 DRAW"S8BM109,155"+N$(S7)
1270 DRAW"S8BM141,155"+N$(S8)
1280 DRAW"S8BM109,181"+N$(S9)
1290 DRAW"S8BM141,181"+N$(S0)
1300 POKE65494,0:GOTO1300
1310 'CHANGE THE WALL
1320 LINE(125,W2)-(127,W3),PRESE
T,B
1330 W2=RND(169)+20:W3=RND(169)+
20
1340 IFW2>W3 THENW4=W2:W2=W3:W3=
W4
1350 W4=W3-W2:IFW4<12ANDW2>40THE
NW2=W2-12ELSEIFW3<160THENW3=W3+1
2
1360 LINE(125,W2)-(127,W3),PSET,
B
1370 W1=0
1380 RETURN
```

FORTH

COME VENTURE FORTH

John Redmond

(John supplied the Forth Compiler in the August Edition of CoCoOz. If you have that tape, load the compiler and then read the article that follows. Perhaps the best illustration of the power of Forth that I can think of is that Forth was the language used to construct Graphicom! G.)

Part 1: Beginning is the hard part.

This article is the start of a series on the Forth language. It will probably turn out to be more than that, though, because I have a horribly pedantic streak. I've had some experience with 8080 and 8086 assembly languages and with Pascal and C; so I suppose I've collected a few insights along the way.

There is no doubt, though, that the most instructive experience that I've had has been the writing of a Forth compiler/interpreter for the Colour Computer. And Almost as instructive was the experience of using the Forth compiler to write a powerful screen editor, an editor than is faster than Wordstar and a lot more logical to use (partly because it doesn't try to be all things to all people). For a job like this, you need a language that can efficiently handle the low-level tasks. Forth can.

It is still a strange beast. I love it and I am infuriated by it, but I never lose respect for its power and scope. To sum up in a sentence: it has the directness of assembly language (and almost as much speed), it is as compact as native machine code and it has control structures equal to any high-level language. Furthermore, because it is both an interpreter and a compiler, it is very simple to debug individual, named, routines from the keyboard. This makes program development much, much easier than with any other language that I know of.

In introductions to the Forth language, it has become customary to make a feature of its peculiarities. Most notable of these are the stack operations. I will depart from this. Until you start to see just what a central interest the stack is to a Forth programmer, the stack 'words' are just a pointless complication.

As we get under way, one necessary warning: Forth thinks backwards - in the sense that it uses postfix (Reverse Polish) notation. To obtain the sum of 4 and 3, you need to enter from the keyboard:

4 3 + . <ENTER>

(Never forget the spaces. If in doubt, put an extra one in.) This will give you the obvious answer on the screen. Now repeat without the full stop. No answer? Where is it? Now type: . <ENTER> and you get your answer. Where was it hiding? On the stack! Now try:

4 3 + <ENTER> 2 + . <ENTER>

Starting to get the idea? The intermediate result (7) was held on the top of the stack and was available for use in another arithmetic operation. In this way, long mathematical expressions can be strung together.

To date, we have used two Forth words. The plus sign (called 'plus') takes the two top values from the stack and repaces them with their sum. In other words, the 4 was put onto the stack, followed by the 3 and these provided the parameters for the addition word. The other Forth word, . ('dot'), is used to print, using the current number base, the value on top of the stack.

To check that you have this clear, type:

4 3 - . <ENTER>

Then use * ('star') or / ('slash') instead of 'minus'. All OK? Or are you surprised at what 'slash' did? Yes, 'slash' is an integer division word. No floating point - the remainder is just thrown away. Arithmetic is simple, isn't it?

Now try:

4 3 mod . <ENTER>

Here you get the remainder after division. With

4 3 /MOD . . <ENTER>
you have two results, the remainder on top of the stack and the quotient under it (so that it's printed second).

Let's focus on the stack for a moment. type:

1 2 3 . . . <ENTER>
and note that the numbers are printed in reverse order. That is because the stack is a LIFO (last-in first-out) or push down structure. It is essential for a Forth programmer to have complete control of the stack. Practise!

Have you made a mistake yet and used the 'dot' word once too often. In such an event, a nonsense value will be printed, together with an error message.

Forth takes its own stack so seriously that checks it for underflow after every single word input from the keyboard - just in case underflow has occurred.

We haven't moved very far yet but, given a handful of new words, we can start doing some useful things. try

100 @ .
(From this point on, it will be understood that every line input from the keyboard is terminated with <ENTER>.) This entry will print the value of the 16-bit word ('cell' in Forth) at memory location 100, while

100 C@ .
prints the byte at location 100. This is analogous to (and much more versatile than) the PEEK function of Basic and far cleaner than the confusing use of pointers in Pascal and C. Analogous to the poke statement of Basic is the ! ('store') word.

33 100 ! and

33 100 C!
store a cell, or a byte, equal to 33 in the current number base, at memory location 100. Try these and check them using the @ and C@ words as above.

I've mentioned number bases a couple of times. Forth is without equal in its ability to alter number bases: Basic, on the other hand, is painful and inconvenient in its use of the &H prefix and the HEX$ function. Try

HEX 100 DECIMAL .
256 is, of course, the decimal equivalent of 100 hex. HEX and DECIMAL are two more predefined Forth words. Interested in binary? Try

: BINARY 2 BASE C! ;
You have just defined a new Forth word which converts the number base to binary. Don't worry about the colon and semicolon. They'll come later. Note that Forth will stay in whatever base you last chose until you change it to something else, such as by using HEX or DECIMAL. Try

BINARY 100 DECIMAL . or

DECIMAL 100 BINARY .

HEX
(Incidentally, if you do something silly, and are answered with an error message, the base will automatically revert to HEX.)

Next month, we will start to look at how to define variable and constants, how to carry out loops and how to use structured control like:

IF .... ELSE .... THEN

BEGIN .... UNTIL
You will find that, given a firm grasp of the basics, you will progress to a level of control which is achieved by only the most dedicated assembly programmers and which is hardly ever thought about by Basic programmers.

To keep you on the straight and narrow, I leave you with one or two useful Forth words to practise with. If you have a Forth Compiler available (a slightly incomplete, but powerful, one was included in a recent CoCoOz), type in:

: BDUMP OVER + SWAP

DO I C@ . LOOP ; <ENTER>
This word which you have defined will display the byte contents of a series of memory locations and requires that you first enter onto the stack the starting address, followed by the number of bytes to be displayed, e.g.,

HEX 200 30 BDUMP <ENTER>

The opposite sort of function can be achieved by the following:

HEX 200 30 3 FILL <ENTER>
This will fill 30 HEX locations, starting at 200 HEX, with the byte value of 3.
Try it, then use the BDUMP word to check that FILL has worked. Experiment, for instance by filling part of video memory starting at 400 HEX. Remember, though, that Forth sits in memory somewhere (the CoCoOz version is between 2000 and 4000 HEX) and that fooling around with this program memory will eventually lead to a total crash. But so what? Switch off, count to ten and reload. It's a learning experience.

3D tic tac toe

16K ECB.

The Delbourgos have supplied an interesting pair of games. They are in fact, the same game; however one is for CoCo, the other is for GoCo and appears in this month's RAINBOW.

Much has been said of this family, most of it true! What needs to be said is that, both in terms of quantity, and quality, over the years, they have contributed a very great deal to our computer's world.

Which is why I am nominating them for the Greg Wilson award at CoCoConf.

Graham.

****

Two-dimensional 3x3 TIC TAC TOE is dull, flat, "plane boring" one might say. On the other hand the 3D version is intriguing, deep, "solidly exciting". Here is a high resolution adaption of the game in which you can pit your wits against CoCo or another player. The pieces are 'balls' and come in cyan and magenta; they may be placed anywhere on the 4X4X4 grid by using the arrow or <,> keys and pressing <enter> when ready to play. A winning line is discerned by the computer at which the play ends and a new game is restarted by pressing <enter>. CoCo plays a decent game but lacks a certain sense of strategy, I must confess; however it will thwart you whenever it percieves an immediate crisis and is quite unforgiving if you make a careless mistake. So don't take it for granted!

The program fits nicely in 16K ECB. As there are no REM lines here are a few remarks about the variables in the listing:

The arrays A,X,Y mark the positions (A=0 if unoccupied, 1 or 5 if filled depending on who has played), and x- and y- coordinates respectively.
M counts the number of pieces successfully entered on the cube (Q=1 if the move was possible, otherwise Q=0).
Entries along the principle diagonals are counted by OO,OX,OY,OZ while the plane diagonals are specified by the arrays DX,XD,DY,YD,DZ,ZD.
The entries along the 16X3 lines parallel to the axes are held in the arrays YZ,ZX,XY.
Subroutine 500-580 carries out the actual assesment of the state of play. With that knowledge the computer finds out where to play in lines 590-740 and then acts upon it (Lines 750-1000).

I won't give anymore away about how CoCo will react to your last move; see if you can figure out how to outsmart it!

The Listing:

```
10 CLS:PRINT"  TIC-TAC-TOE IN 3
dimensions":PRINTSTRING$(32,198)
;:PRINT"          BY BOB DELBOURGO
":PRINTSTRING$(32,198);
20 PRINT"USE ARROW KEYS AND >,<
KEYS TO  MOVE THE ORANGE CURSOR
ALONG THE3 ORTHOGONAL DIRECTIONS
. WHEN   READY TO INSERT PIECE,
<ENTER>."
30 PRINTSTRING$(32,198);:PRINT"T
HE FIRST PLAYER TO PLACE  FOUR P
IECES OF THE SAME COLOR ALONG AL
INE, EITHER PARALLEL TO ONE OF T
HE AXES OF THE CUBE OR ALONG   O
NE OF THE DIAGONALS WINS."
40 PRINTSTRING$(32,198);:PRINT@4
80,"WAIT FOR INITIALIZATION...."
;:GOSUB440
50 PRINT:PRINTSTRING$(32,198);:P
RINT" <1> VS.coco, OR <2> VS.PLA
YER?":PRINTSTRING$(32,198);
60 C$=INKEY$:C=VAL(C$):IFC<1ORC>
2THEN60
70 PMODE1,1:PCLS1:R=RND(-TIMER)
80 Q$="C4UBUU2R3U3L6D"
90 CLS5:PRINTSTRING$(32,198);
100 IFC=1THENPRINT"OK, YOU (="CH
R$(223)") GO FIRST,":PRINT"COCO
(="CHR$(239)") GOES NEXT"
110 IFC=2THENPRINT"OK, "CHR$(223
)" PLAYS FIRST, AND THEN "CHR$(2
39)
120 PRINTSTRING$(32,198);
140 COLOR2:FORI=1TO4:FORJ=1TO4:L
INE(X(I,J,1),Y(I,J,1))-(X(I,J,4)
```

```
,Y(I,J,4)),PSET:NEXTJ,I
150 COLOR3:FORI=1TO4:FORK=1TO4:L
INE(X(I,1,K),Y(I,1,K))-(X(I,4,K)
,Y(I,4,K)),PSET:NEXTK,I
160 COLOR3:FORJ=1TO4:FORK=1TO4:L
INE(X(1,J,K),Y(1,J,K))-(X(4,J,K)
,Y(4,J,K)),PSET:NEXTK,J
170 SCREEN1,1:FORI=1TO4:FORJ=1TO
4:FORK=1TO4:FORR=1TO5STEP2:CIRCL
E(X(I,J,K),Y(I,J,K)),R,5:NEXTR:C
IRCLE(X(I,J,K),Y(I,J,K)),6,8:CIR
CLE(X(I,J,K),Y(I,J,K)),8,5:NEXTK
,J,I
180 DRAW"BM210,190;C4U8R8D4L8BE4
BR8D8R8BR4U4E4F4L8R8D4BR8U4H4F4E
4"

190 PCOPY1TO3:PCOPY2TO4:I=RND(4)
:J=RND(4):K=RND(4)
200 P=P+1:IFP=3THENP=1
210 FORH=1TO6:CIRCLE(200,185),H,
C(P):NEXTH
220 IFP=2ANDC=1THEN770
230 CIRCLE(X(I,J,K),Y(I,J,K)),8,
8:PLAY"V4L801C":CIRCLE(X(I,J,K),
Y(I,J,K)),8,5
240 I$=INKEY$:IFI$=""THEN230
250 IFI$=CHR$(8)THENJ=J-1:IFJ<1T
HENJ=1
260 IFI$=CHR$(9)THENJ=J+1:IFJ>4T
HENJ=4
270 IFI$=CHR$(10)THENK=K-1:IFK<1
THENK=1
280 IFI$=CHR$(94)THENK=K+1:IFK>4
THENK=4
290 IFI$=CHR$(44)THENI=I+1:IFI>4
THENI=4
300 IFI$=CHR$(46)THENI=I-1:IFI<1
THENI=1
310 IFI$=CHR$(13)THEN320ELSE230
320 IFA(I,J,K)<>0THENSOUND10,10:
GOTO230
330 FORH=1TO6:CIRCLE(X(I,J,K),Y(
I,J,K)),H,C(P):NEXTH:PLAY"L255V3
005GECGECGECGEC":A(I,J,K)=N(P):M
=M+1:Q=1
340 GOSUB1010:IFM<7THEN200
350 GOSUB500:N1=4:GOSUB590:N1=20
:GOSUB590:IFN0=40RN0=20THEN380EL
SEIFM=64THEN470
360 GOSUB1010
370 Q=0:GOTO200
380 LINE(208,180)-(255,191),PRES
ET,BF:GOSUB1010
390 DRAW"BM208,180C4D8E4F4U8BR4D
8BR4U8F8U8BR12L8D4R8D4L8":FORH=1
TO4:PLAY"L255V3005BAGFEDC04BAGFE
DC03BAGFEDC02BAGFEDC01BAGFEDC":N
EXTH400 I$=INKEY$:DRAW"BM4,8;C6H
4E4BR12L8D4R4L4D4R8BR2U8F8U8BR2R
8L4D8BR14L8U4R4L4U4R8BR2BD8U8R8D
4L8R4F4BR4E4H4"
410 DRAW"BM4,8;C7H4E4BR12L8D4R4L
4D4R8BR2U8F8U8BR2R8L4D8BR14L8U4R
4L4U4R8BR2BD8U8R8D4L8R4F4BR4E4H4
"
420 IFI$=CHR$(13)THENPCOPY3TO1:P
COPY4TO2:P=0:FORI=1TO4:FORJ=1TO4
:FORK=1TO4:A(I,J,K)=0:NEXTK,J,I:
M=0:Q=0:N=0:N1=0:N0=0:GOTO190
430 GOTO400
440 C(1)=6:C(2)=7:N(1)=5:N(2)=1:
DIMA(4,4,4),X(4,4,4),Y(4,4,4),YZ
(4,4),ZX(4,4),XY(4,4),DX(4),XD(4
),DY(4),YD(4),DZ(4),ZD(4)
450 FORI=1TO4:FORJ=1TO4:FORK=1TO
4:A(I,J,K)=0:X(I,J,K)=106+47*(J-
1)-33*(I-1):Y(I,J,K)=152-48*(K-1
)+11*(I-1):NEXTK,J,I
460 P=0:RETURN
470 LINE(0,0)-(96,8),PRESET,BF:L
INE(188,180)-(255,191),PRESET,BF
480 DRAW"BM190,190;C4U4E4F4L8R8D
4BR6U8R4F4G4L4BR10U8R8D4L8R4F4BR
2U4E4F4L8R8D4BR2U8D8E4F4U8"
490 GOTO400
500 OO=0:OX=0:OY=0:OZ=0:FORI=1TO
4:DX(I)=0:DY(I)=0:DZ(I)=0:XD(I)=
0:YD(I)=0:ZD(I)=0:FORJ=1TO4:XY(I
,J)=0:YZ(I,J)=0:ZX(I,J)=0:NEXTJ,
I
510 IFQ=1THENDRAW"BM0,8C4U4E4F4L
8R8D4BR4R8U4L8U4R8BR12L8D4R8D4L8
BR18L4U4R2L2U4R4BR12L8D4R8D4L8BR
12R8U4L8U4R8BR6D8BR6U8F8U8BR10L6
D8R6U4L2"
520 FORU=1TO4:FORV=1TO4:FORW=1TO
4

530 YZ(U,V)=YZ(U,V)+A(W,U,V):ZX(
U,V)=ZX(U,V)+A(V,W,U):XY(U,V)=XY
(U,V)+A(U,V,W):NEXTW
540 DX(U)=DX(U)+A(U,V,V):XD(U)=X
D(U)+A(U,V,5-V)
550 DY(U)=DY(U)+A(V,U,V):YD(U)=Y
D(U)+A(V,U,5-V)
560 DZ(U)=DZ(U)+A(V,V,U):ZD(U)=Z
D(U)+A(V,5-V,U):NEXTV
570 OO=OO+A(U,U,U):OX=OX+A(5-U,U
,U):OY=OY+A(U,5-U,U):OZ=OZ+A(U,U
,5-U):NEXTU
580 RETURN
590 N=0:O=0:XO=0:YO=0:ZO=0:I0=0:
J0=0:FORI=1TO4:IFDX(I)=N1 THENN=
DX(I):O=I
600 IFDY(I)=N1 THENN=DY(I):O=4+I
610 IFDZ(I)=N1 THENN=DZ(I):O=8+I
620 IFXD(I)=N1 THENN=XD(I):O=12+
I
630 IFYD(I)=N1 THENN=YD(I):O=16+
I
640 IFZD(I)=N1 THENN=ZD(I):O=20+
```

```
I
650 NEXTI:IFOO=N1 THENN=OO:O=25
660 IFOX=N1 THENN=OX:O=26
670 IFOY=N1 THENN=OY:O=27
680 IFOZ=N1 THENN=OZ:O=28
690 FORI=1TO4:FORJ=1TO4
700 IFYZ(I,J)=N1 THENN=YZ(I,J):I
0=I:J0=J:XO=1
710 IFZX(I,J)=N1 THENN=ZX(I,J):I
0=I:J0=J:YO=1
720 IFXY(I,J)=N1 THENN=XY(I,J):I
0=I:J0=J:ZO=1
730 NEXTJ,I:IFN<>0THENN0=N
740 II=0:I=I-1:J=J-1:Q=0:RETURN
750 IFA(R,S,T)=0THENPLAY"L255V30
02GECGECGECGEC":FORH=1TO6:CIRCLE
(X(R,S,T),Y(R,S,T)),H,C(P):CIRCL
E(X(R,S,T),Y(R,S,T)),H,8:CIRCLE(
X(R,S,T),Y(R,S,T)),H,C(P):NEXTH:
PLAY"L255V3002GECGECGECGEC":A(R,
S,T)=N(P):M=M+1:Q=1:RETURN
760 Q=0:RETURN
770 DRAW"BM6,8;XQ$;BM18,8;XQ$;BM
30,8;XQ$;":Q=0:GOSUB500
780 IFM<5THENR=1+RND(2):S=1+RND(
2):T=1+RND(2):GOSUB750:IFQ=0THEN
780ELSEI=I-1:J=J-1:GOTO340
790 N1=3:GOSUB590:IFN=3THEN840
800 N1=15:GOSUB590:IFN=15THEN840
810 N1=2:GOSUB590:IFN=2THEN840
820 N1=10:GOSUB590:IFN=10THEN840
830 R=RND(4):S=RND(4):T=RND(4):G
OSUB750:IFQ=0THEN830ELSEI=I-1:J=
J-1:GOTO340
840 IFO>0ANDO<29 THENONO GOTO880
,880,880,880,890,890,890,890,900
,900,900,900,910,910,910,910,920
,920,920,920,930,930,930,930,940
,950,960,970
850 IFXO=1THEN980
860 IFYO=1THEN990
870 IFZO=1THEN1000
880 FORII=1TO4:R=0:S=II:T=II:GOS
UB750:IFQ=1THEN340ELSENEXTII
890 FORII=1TO4:R=II:S=0-4:T=II:G
OSUB750:IFQ=1THEN340ELSENEXTII
900 FORII=1TO4:R=II:S=II:T=0-8:G
OSUB750:IFQ=1THEN340ELSENEXTII
910 FORII=1TO4:R=0-12:S=II:T=5-I
I:GOSUB750:IFQ=1THEN340ELSENEXTI
I
920 FORII=1TO4:R=II:S=0-16:T=5-I
I:GOSUB750:IFQ=1THEN340ELSENEXTI
I
930 FORII=1TO4:R=II:S=5-II:T=0-2
0:GOSUB750:IFQ=1THEN340ELSENEXTI
I
940 FORII=1TO4:R=II:S=II:T=II:GO
SUB750:IFQ=1THEN340ELSENEXTII
950 FORII=1TO4:R=5-II:S=II:T=II:
GOSUB750:IFQ=1THEN340ELSENEXTII
960 FORII=1TO4:R=II:S=5-II:T=II:
GOSUB750:IFQ=1THEN340ELSENEXTII
970 FORII=1TO4:R=II:S=II:T=5-II:
GOSUB750:IFQ=1THEN340ELSENEXTII
980 FORII=1TO4:R=II:S=I0:T=J0:GO
SUB750:IFQ=1THEN340ELSENEXTII
990 FORII=1TO4:R=J0:S=II:T=I0:GO
SUB750:IFQ=1THEN340ELSENEXTII
1000 FORII=1TO4:R=I0:S=J0:T=II:G
OSUB750:IFQ=1THEN340ELSENEXTII
1010 LINE(0,0)-(96,8),PRESET,BF:
RETURN
```

size down is 124 chars. and unless you need 250 then use this size. Put you key info in the primary and use the secondary to expand the database as required. To check the size, count the size of the individual fields in the primary and secondary formats and make sure they are both under 124 or 250 as the case may be.

Now you have created the dictionary file, it is time to create the database. Type RMSNEW (filename), where filename is the same prefix name as the dictionary file, and you will be prompted for record size and number of records. For this example use 124 for size and 40 for number of records. On a double sided 40 track drive you could have a database of up to 2800 records if you use the disk as a data disk and have the RMS control files on the system disk. This would require two drives, but then if you are using FLEX, you should have two drives anyway as things get messy with only one.

This should get you started. I haven't mentioned control keys as different versions of RMS utilize different keys. Check your manual for details. When you have done all this , type RMS (filename) and the screen fill in form should appear. If you get an error, go back and check the dictionary file for mistakes.

Play around with entering and finding records, and next month I will present a report file to go with your database. See you then.....

FANTASTIC FLEX

OZY O9
(2)

by Bob Thompson

Well hi again. This time I will look at the Shell (and hopefully get it right).

The Shell holds nothing to fear as neither do the BASIC ROM's. This is because they do the similar function,of interpreting a command and acting upon it.

In BASIC you get OK in os9 you get OS9:. However the Shell has a few differences (good ones). One of these is the ability to read a file as a command.

Using the startup file on the system disk we will have a look at some of the things it can do.(use a backup!!!!).

First off a bit of house keeping. Remember these are sugestions only. All my text files have headings in lower case, directory's are all upper case and programs used by other programs have three uppercase and the balance in lower case. This way I don't try to execute a text file or list a directory,it also starts to make things easer to find.

The second part is keeping similar units together. All my Basic09 source files are held under a directory called BASIC09.SRC, and machine code source under ASM.SRC. These are uncompiled and can be edited. More on this another time

Type :LIST STARTUP <enter>

This will show the startup file. While I think of it did you like the way the Shell ignores the difference between upper and lower case NO! sorry bout that.

Type :list startup <enter>

How did I get to lowercase (shift 0). Didn't work hey HHHMMMMM!!!!

Type :TMODE /1 -UPC

Now shift 0 and it should work ??

Anyway to continue, what if you want a hardcopy (that's tech for a copy that's hard to get) no it's not just pulling your joystick. Send to printer.

Type :LIST STARTUP >/P

This means list the file (startup) and send the output '>' to the printer '/P'.

Great now let's send it to another file (possibly to modify without hurting the original). It must have a new name.

Type :LIST STARTUP >/D0/MYNEWFILE

Now list MYNEWFILE and hopefully it will be the same.

Some of the other things burried in the manual are :DIR X ,this will show you the current execution directory wherever it is .Instead of :DIR /D?/CMDS

You may also look at it's most secret workings :DIR X E or :DIR E.

The > , < , are used to show paths for output and input . We used the ouput for sending a file to the printer,but what about input ??. Maybe next time we'll talk about boots and using SDISK for double sided drives and 40 tracks, stepping rates of 6ms etc.

So for now BOB T (X MODE is not for over 16's).....

stepping out with McINTOSH

By A.L.McIntosh

When I first got my hands on the OS-9 operating system, my first thoughts were that it had to be powerful.., Just look at all the Documentation there is with it. So as I sat down with my two double sided M.P.I Drives I couldnt help wondering what was missing. The system only caters for a single sided Tandy drive with 35 tracks and a track to track stepping rate of 30 milliseconds. My M.P.I's were being grosely under-used. Surely there must be a way to use both sides of my drives, all 40 tracks and jump the stepping rate up to a respectable 6 milliseconds. So I set to work to make this possible. I remembered reading in a copy of RAINBOW that there was a way to do it so that it booted up to use all these features. Unfortunately, I was unable to make too much sense of the article. Although I did manager to gleen some very usefull information from it.

There are a number of modules, or short programs which the OS-9 Kernel uses for disk input/output functions. These are called D0,D1,D2,D3. There is then the main driver for the whole of the disk system called CCDISK. My interpretation of what happens when you use a disk command are as follows. I cant say if its 100% correct but it will give you an idea of what goes on. If you issue a command to OS-9, It first looks in its memory for the program you called.(All commands are only short programs, which do specific duties.) In these programs are values and tables which are used to talk to the disk. One of these would be equal to the number of tracks available to the disk to read or write to. Now if this number was changed, then we could nominate how many tracks were on each disk. All the parameters for disk control are contained somewhere inside these programs, or modules. My only problem was how to change them. OS-9 also has a utility program called DEBUG. This program has the ability to load a program, examine, and change any location you choose. Now after many phone calls and plenty of pulling of hair, I thought why not put all these together. When OS-9 boots up it looks for a file called of all things 'STARTUP' from where it learns what to do to begin a work session. If you were to use DEBUG to call up and load the device descriptors (D0,D1,D2,D3) then it should be able to change any locations. ....At last a light at the tunnel. Using the addresses and clues contained in the

rather cryptic article in Rainbow, I came up with a way of booting the operating system so that when it starts you have 40tracks per drive, you can use both sides of the drive, with one being Drive 0 and the other drive 2 while on the other disk its drive 1 and drive 3. And to put the icing on the cake, it also gives you a 6 millisecond stepping rate. First it loads the CCDISK module and carries out a few changes on it. Then its the device descriptors, D0-D3. After that it prints out a text file which tells the user what is happening.

Here is a listing of the file you have to build. And what you have to do to achieve the desired result. The first thing you have to do is to kill off the existing startup file. type DEL STARTUP then type BUILD STARTUP The BUILD command will come back with a two question mark prompt, then type

?? DEBUG </d0/startup2 <hit enter>
?? <hit enter>

This will be the file that is loaded by the debuger.

```
$hires              Loads hires module
                    from OPACK
$tmode -upc         goes into lower case
L CCDISK            Loads the module
                    CCDISK
 . .+1FE            move forward 1FE
                    bytes
=10                 changes current byte
                    to 10
L CCDISK            This is the same for
                    all the
. .+204             next lines until we
                    load
=08                 the ECHO module
=8b                      "      "
L CCDISK                 "      "
. .+2DD                  "      "
=40                      "      "
L CCDISK                 "      "
. .+2E9                  "      "
=00                      "      "
L CCDISK                 "      "
. .+210                  "      "
=41                      "      "
=42                      "      "
L D0                Now we load the disk
                     drive
. .+18              descriptors, and make
=28                 some changes in these
L D1
. .+18
=28
L D2
. .+18
=28
L D3
. .+18
=28
$LOAD echo
$echo Drives now set to 6 millisecond
step
$echo   drives also have 40 tracks
available
$echo  Double sided drives will have
D0,D1,D2,D3
$echo   set
$echo
$printerr
$echo   The error printing routine is
now also active.
$xmode /p baud=3
$echo Printer rate set to 1200 baud.
Q                   The 'Q' terminates the
                    file
```

The $LOAD echo command loads a command module that will echo to the output path, in this case the screen, whatever is typed on the same line after it. So we can put a little bit of information about our changes there. That is all there is to it.

This is about the easiest way I know how to go about doing the changes, I know there is a better way but I haven't worked it out yet. If anyone has done it and gets the same results please let us all know about it.

Without going into all the details at this point. Rather than put all of the above into the STARTUP file, why not build a temporary proceedure file exactly as above to change the device descriptors D0,D1,D2,D3 and CCDISK. Call the proceedure file to effect these changes. Then create a new SYSTEM disk using COBBLER and then DSAVE the CMDS file and any other files you want on your SYSTEM disk. You will then have a SYSTEM disk that will boot with all these changes intact and no need for such a long STARTUP file. I know DSAVE is a difficult command to grasp but it is well worth the study and effort. Perhaps we will dedicate an article to that command in the near future.

Kevin.

A LOOK AT OS-9

We will take a short look at some of the commands and try to clear a couple of the most common problems.

The very first thing that you should do when you open the package is to write protect the two disks.

The next thing is backup the OS-9 BOOT DISK using the DISK BASIC command BACKUP, if you have only one drive just type BACKUP 0. This command is not mentioned in the BASIC manual.

OS-9 must be loaded from drive 0, so insert the newly backed up disk in your drive and type RUN"*. If your drives are working OK under Disk Basic then don't worry about the speed check program (it always tells me that all my drives are slow).

After you have answered the date prompt you will be in the SHELL. The next thing to do is to make a backup copy of your system disk (all this is also in the Getting Started Manual).

The first command and one of the most used is the DIR command. Many people have trouble with the hierarchial directory structure used by OS-9.

One of the problems most encountered is when you change disks and every time you type a command you encounter an error. One of the most likely causes for this is that you changed the disk and forgot to tell the DOS about it. There are two commands that handle this.

The first thing OS-9 does is assign /d0/cmds as your system directory, that's the one that all your commands are stored in. It also assigns /d0 as your data directory, that's the working directory which is where your text files, data and other files are stored. There will be other directories but we won't look at that yet.

There are two commands that are used when you change disks they are CHD and CHX. CHD is CHange Data directory and CHX is CHange eXecution directory. For example: To link BASIC09 after swapping your BASIC09 disk for your SYSTEM disk then it is necessary to CHX /D0 followed by typing the command BASIC09.

The big news on the OS-9 front at the moment is the USER GROUPS and the public domain software available through the group. The address for the user group is P.O.Box 8027, Des Moines, IA 50301 .All they ask for dues is $25US which I think is quite reasonable.

They publish a newsletter called MOTD. So far they have released 6 disks of programs and utilities, some are not usable unless you have a double sided 80 track drive.

I have all the software except the two that require the 80 track drives, those haven't been sent yet but all the others are available.

The idea behind the public domain library is that you get one free disk when you join. This disk is sent by Frank Hogg Labs in the US, this also puts you on his maining list for a small magazine that he publishes which tells you about all the latest programs he puts out and he has also hired some of the experts to write columns for it.

To get the other disks either send $3 for each additional disk or send some of your own original programs or non-trivial alterations to their programs. You will get a free disk for each program that you send. If you wish the spelling checker and dictionary (disk #1 & #2) you need to tell them that you have 80 track drives.

The purpose of this column is to help you to understand and use OS-9, so if you have any questions about OS-9 write to me, but first READ THE MANUALS FROM COVER TO COVER (perhaps starting with the red Commands Manual).

Jack Fricker.
P.O.Box 306.
Clayfield 401

MiCO

I must confess to some suprise, and pleasure at the number of new CoCo users contacting us asking for help in getting the listings they have typed from MiCo to run. It is probably well known among "older hands" but MiCo BASIC programs will generally run with only minor modification in CoCo.

When people tell you why they buy a home computer, somewhere in their answer you will usually hear the word 'education'. With computers, education can perhaps be broken into two further divisions, traditional education and the learning about computers. Much has been said in so many different ways about the use of computers in traditional fields of education and yet I believe learning about the dreaded machines themselves is in itself a worthwhile end. How many of those who purchase a home computer can see a clearly defined application for their purchase? Even if you do know what you plan to use your computer for then I bet you had the greatest difficulty in selecting your machine. The old story of too many compromises to be made.

The truth is computers have not yet found a place in the home that meets everyone's vison of the future. Until that happpens, and it will happen, it will take some time for you to establish a purpose for your computer in your daily life. In the meantime why not make a little effort and learn about the machine itself. BASIC is a superb language for knocking up those short programs to perform specific tasks. By simply typing in listings and getting them to run, you will become familiar with the way the language works and begin to see some of the ways other people meet their own needs. In no time you will begin to develop your applications and eventually you will be able sit back and say that your computer has become a useful tool in your own daily life.

The much maligned phrase 'Computer Literacy' is all about this developing an understanding of just what you can sensibly do with computers and those tasks you personally are better to perform the good old fashion manual way. Some people can balance their checkbook in their head and it would perhaps be senseless for them to carry out that task on a computer, and yet for others a simple chequebook balancing program makes a formidable monthly task nothing more than a simple chore.

So sit down and play with your machine, type in a few listings, make them run and then write a few of your own programs. In no time at all you will find that you too will develop a feel for the place a computer may take in your own life.

And what about MiCo? MiCo BASIC is an ideal medium for the development of this computer literacy. Generally the listings are easy to follow and do not demand that you learn a heap of fancy commands. So those CoCo users who are only now discovering what the MiCo team have known all along, don't be afraid to type in these listings. There are not too many that will not work on your machine. All of this goes double for MiCo users, after all we are already in the box seat.

LETTERS

Dear Graham,

Thank you for sending me a copy of your magazine. Unfortunately I am more interested in the internals of the machine.

For instance, I am interested in a program that converts the MC-10 into a serial terminal. For that purpose I require a disassembled and commented listing of BASIC. This is necessary, because it appears that one of the parallel port lines is used to effect the output of the serial I/O line.

Since you informed me that such a listing is not available I have written a disassembler to do the job. In appreciation of your sending me a copy of your magazine I am sending you a copy of this program. I have also included some documentation. Writing down this documentation has in fact slowed down my reply to you.

There are some other possibilities in the MC-10 that interest me. For instance I have some software that I may convert for use on the MC-10. This includes a relocator for machine code programs, Forth and an assembler. Unfortunately many of these require a reasonable amount of memory.

I have obtained some photocopies of articles about the MC-10 from David Baker. These were published in CoCo and Rainbow. From these it is clear that the graphics can be extended to 192*256. I have recently performed the memory modifications required and it appears that all the Color Computer graphics modes are available. It was not too easy, since the article describing the mods had a number of errors and also related to the U.S version. The memory is now expanded to 8 kilobytes. All this is available to BASIC. This in itself is a problem since both the text screen and the BASIC program itself occupy the top half of the HI RES graphics screen. At this time I have not devised a method of shifting BASIC into the undisplayed 2 kilobyte part of new memory.

I have assigned the contacts on the edge connector and I should have no trouble in extending the memory to 64K. I intend to do this by using the bus system devised by Elektor magazine. In this way a number of peripherals including a disc drive controller will be available to me. Silly isn't it.

I have an additional question. How can I obtain a copy of the MC-10 circuit diagram or even better a technical manual. I have been pestering my local Tandy man for three months now with no joy. I really have no idea how some of your readers get so enthusiastic over Tandy products, since this type of thing appears to be commonplace.

Anyhow, in appreciation of your sending me an inspection copy of your magazine, I am returning the magazine and a copy of a DISASSEMBLER written in BASIC.

R.G. Rothwell
PRESTON. Vic.

We have one Reg Lang here on the Gold Coast who is in the throes of producing a speech pack for the MC 10. The next project will be a speech recognition unit! I heard of another lad the other day who has a dedicated MC 10 to control his home's heating system. The MC 10 is not a dead end computer!

Graham.

To Mico

Recently I discovered a way to draw circles on the MiCo. The following listing should do the trick:

```
10 CLS:A=22/7:SI=12:SH=9:H=30:V=
   15
20 FOR E=0TO360
30 X=COS(A/180*E)*SI+H
40 Y=SIN(A/180*E)*SH+V
45 SET(X,Y,5)
50 NEXTE
```

This listing is only an example. You may change the SI and SH values to get squashed up or fat circles, but to fit the circle in from the center of the screen the SI values can't go over 15. If you want to speed up the drawing of the circle, on line 20, step E, 3 or 4 times just so the MC-10 doesn't plot all the points of the circle.

I have also wondered how to draw a line between two points. The best way I can do this is shown in the following listing. The listing, although it does work, it sometimes has bugs.

```
10 X=10:Y=11:X1=20:Y1=19:CLS0
15 M=(Y1-Y)/(X1-X)
20 FOR H=XTOX1STEP.25
25 Y=M*H
30 X=Y/M
35 SET(X,T,5)
40 NEXTH.
```

This program works well as long as the gradient (M value) does not pass 3 or as long as the line isn't an almost vertical line. This is where I want help. Obviously you can change the X,Y and X1,Y1 values because these are the points you are drawing between. Otherwise it performs alright.

N.B. Your X value is always lower than your X1 value. It does not matter where they are on the y-axis.

Shane Herbert.

Dear Graham,

I have just typed in "Skier" in this month's MICO and find that the program seems to hang up. When I run the program I only get a series of lines down the screen and no commands seem to work, not even BREAK, and it takes about six resets to call back the program. I thought it was me so I typed the whole program out again but got the same effect. I wonder if you could please give me some advice as to how I could get this program to run properly.

K. Gurnhill
LOFTUS. N.S.W.

Dear K...

We'll ask the author! Are you sure you didn't have OS8 in the computer at the time? That would almost certainly create at least some of the difficulties that you describe!

Seriously, we do check each program before it goes into the magazine. Generally, the Authors supply the program on a tape, so it is easy to run it to see what it does. From there it gets listed, and you receive that listing in the magazine.

There have been exceptions to this rule, particularly in the early days, where we trusted that the Author had checked his program. I have a horrible feeling that SKIER came under this heading!

We'll ask the Author!
Graham.

Dear Graham,
I'd like to say how much I enjoy the magazine. The new format is easy to read and the listings clear and interesting. Combining the CoCo and MiCo magazines was a great idea and means that my learning is not restricted to the limitations of the MC 10.

As a virtual novice to coding I find it very frustrating when confronted with a listing I can't follow because it uses POKES I don't understand. I know they need to be used on occasions for speed or to achieve some effect that can't be called from BASIC, but please, include some sort of brief explaination of what they are there for. How else am I going to learn?

This is doubly so when the listing is not for the MiCo. There is nothing more frustrating than typing in a program only to find it generates rubbish (or worse, nothing at all) because the memory addresses used are for another machine and there is no indication as to how to modify it for the MiCo.

After talking with several other people who have bought MiCo's, I think that at least one good explanatory article on some aspect of programing / coding in each issue would be well received. The writings of Tom Lehane, Rod Hoskinson, and Graham Pollock, amoungst others, helped me to understand the basics in a way that was clear and sensible, and would bear reprinting for those who missed them in the early days. I find Michael Turk's articles confusing, but that's probably because he is further down the track than I.

The Central Coast (Gosford) Meet has now been changed to the third Tuesday of the month at Gosford Primary School. No doubt John Morton-Jones or Peter Seifert will let you know further details soon. We have started a MiCo group that meets irregularly, though we are in constant contact by phone. The local Tandy store has sold over 100 MiCo's in the last few months - my information is that they are no longer available at any price - so there are obviously a few people out there who could benefit from getting together.

BRUTE FORCE

USER NOTES
Written by Richard Rothwell

Theory of Operation
****

This disassembler is written in BASIC. It converts 6800/6801/6803 machine language opcodes into the corresponding mnemonics. This makes it possible to identify the purpose of the many machine code subroutines that make up the BASIC interpreter. In turn, this allows faster BASIC programs to be written.

This disassembler uses a very simple "table look-up" approach. This means that the logic of the program is very simple, the BASIC program is very small and the program is very fast. The main speed limitation is the speed of your printer. The price paid for this is a very large data table. Furthermore by changing the data table the same program can be used for other microprocessors.

Format of the Data Table
****

The data table is shown in listing 1. Note that the mnemonics are listed in order of their opcodes. Spaces in the table represent opcodes which have no function in the 6803 microcomputer chip. The mnemonic for each opcode is stored in memory as a 6 byte string. The first 5 bytes contain the actual mnemonic plus spaces for padding. The 6th byte is an ASC11 character, representing the number of data bytes that follow the mnemonic. Listing 1 also shows the memory address for the first byte of each mnemonic.

Installing the Data Table
****

The first task in installing the disassembler is to store the data table in memory. The table is then checked. Then the table is stored on tape.

These tasks are accomplished by using the program in listing 2. Type this listing into the MC-10 then save it on tape. This is actually 3 programs but it is convenient to store them on tape together.

The data table is entered from the keyboard and stored in memory by typing "RUN 1000". The program then prompts the user for the required data. Take special care that 6 characters are typed always. This includes spaces. To input a blank entry simply type (return). This program can also be used to correct mistakes.

The table is checked by typing "RUN 3000". The program will then print out the entire list. If a mistake has been made, identify the address of the faulty mnemonic, then use the "RUN 1000" program to correct the error.

The data table is stored on tape using that part of the program beginning at line 2000. However special comments need to be made about this program. The data table will be stored on tape as a binary file. This is a very efficient format for storing data. However this function is missing from the MC-10's BASIC interpreter. This function is called CSAVEM and it can be used on the Color Computer. The program that simulates this function was adapted from H.Allen Curtis' article in the February 1984 issue of the "Rainbow".

You should have already typed in the entire BASIC program. So run 2000. Break out of the question mark prompt. List the program and you will notice that string A$ has been messed up. Don't worry, the machine language program listed in the data table has now been embedded in the string. Now delete lines 2050-2080 and 2100-2180 inclusive. The entire BASIC program can now be saved again on tape. This is now your working copy.

To save the data table use this CSAVEM" program by typing "RUN2000". A question

mark will appear. Enter the data for the file in the following way: 18944,20479,"MNEMTABL". Start your tape recorder in record mode, then press return. When recording is finished the OK prompt will reappear.

Now that's all done give yourself a break and grab a cup of coffee.

Installing the BASIC program

****

This is the easy part. The program is shown in listing 3. It is a straight forward program that is typed in from the keyboard. CSAVE it.

Using the Brute Force Disassembler

****

Now is the time to put it all together. CLOAD the BASIC program. Check that it loaded correctly. Then CLEAR 100,18944 to protect the data table. CLOADM the data table. The M is not a typo. It is an undocumented variation of the CLOAD command that loads a binary file or machine code program.

As a test run the program. In response to the prompts enter consecutively: 63278(return), 255(return), RTS (return). For these input values the program will begin to disassemble at address 63278 and finish disassembly after 255 bytes or when it reaches a return from subroutine op-code. For your information 63278 is the starting point for the routine that executes whenever the MC-10 is powered up or reset. Some sample output is included in listing 4.

If you have not got a printer change LPRINT to PRINT and insert a STOP command at line 1175. Thus you can record each line and proceed by typing CONT.

Be careful when using the program. The starting address should be the starting point of a known program or subroutine. The JSR, BSR, and JMP instructions are good ways of identifying the beginning of a code segment. Attempting to disassemble a ROM data table will only produce garbage.

DECEMBER, 1984.

LISTING 1

****

```
 18944
 18950 NOP  0
 18956
 18962
 18968 LSRD 0
 18974 ASLD 0
 18980 TAP  0
 18986 TPA  0
 18992 INX  0
 18998 DEX  0
 19004 CLV  0
 19010 SEV  0
 19016 CLC  0
 19022 SEC  0
 19028 CLI  0
 19034 SEI  0
 19040 SBA  0
 19046 CBA  0
 19052
 19058
 19064
 19070
 19076 TAB  0
 19082 TBA  0
 19088
 19094 DAA  0
 19100
 19106 ABA  0
 19112
 19118
 19124
 19130
 19136 BRA  1
 19142
 19148 BHI  1
 19154 BLS  1
 19160 BCC  1
 19166 BCS  1
 19172 BNE  1
 19178 BEQ  1
 19184 BVC  1
 19190 BVS  1
 19196 BPL  1
 19202 BMI  1
 19208 BGE  1
 19214 BLT  1
 19220 BGT  1
 19226 BLE  1
 19232 TSX  0
 19238 INS  0
 19244 PULA 0
 19250 PULB 0
 19256 DES  0
 19262 TXS  0
 19268 PSHA 0
 19274 PSHB 0
 19280 PULX 0
 19286 RTS  0
 19292 ABX  0
 19298 RTI  0
 19304 PSHX 0
 19310 MUL  0
 19316 WAI  0
 19322 SWI  0
 19328 NEGA 0
 19334
 19340
 19346 COMA 0
 19352 LSRA 0
 19358
 19364 RORA 0
 19370 ASRA 0
 19376 ASLA 0
 19382 ROLA 0
 19388 DECA 0
 19394
 19400 INCA 0
 19406 TSTA 0
 19412
 19418 CLRA 0
 19424 NEGB 0
 19430
 19436
 19442 COMB 0
 19448 LSRB 0
 19454
 19460 RORB 0
 19466 ASRB 0
 19472 ASLB 0
 19478 ROLB 0
 19484 DECB 0
 19490
 19496 INCB 0
 19502 TSTB 0
 19508
 19514 CLRB 0
 19520 NEG .1
 19526
 19532
 19538 COM .1
 19544 LSR .1
 19550
 19556 ROR .1
```

BRUTE FORCE

```
19562 ASR .1
19568 ASL .1
19574 ROL .1
19580 DEC .1
19586
19592 INC .1
19598 TST .1
19604 JMP .1
19610 CLR .1
19616 NEG  2
19622
19628
19634 COM  2
19640 LSR  2
19646
19652 ROR  2
19658 ASR  2
19664 ASL  2
19670 ROL  2
19676 DEC  2
19682
19688 INC  2
19694 TST  2
19700 JMP  2
19706 CLR  2
19712 SUBA#1
19718 CMPA#1
19724 SBCA#1
19730 SUBD#2
19736 ANDA#1
19742 BITA#1
19748 LDAA#1
19754
19760 EORA#1
19766 ADCA#1
19772 ORAA#1
19778 ADDA#1
19784 CPX #2
19790 BSR  1
19796 LDS #2
19802
19808 SUBA 1
19814 CMPA 1
19820 SBCA 1
19826 SUBD 1
19832 ANDA 1
19838 BITA 1
19844 LDAA 1
19850 STAA 1
19856 EORA 1
19862 ADCA 1
19868 ORAA 1
19874 ADDA 1
19880 CPX  1
19886 JSR  1
19892 LDS  1
19898 STS  1
19904 SUBA.1
19910 CMPA.1
19916 SBCA.1
19922 SUBD.1
19928 ANDA.1
19934 BITA.1
19940 LDAA.1
19946 STAA.1
19952 EORA.1
19958 ADCA.1
19964 ORAA.1
19970 ADDA.1
19976 CPX .1
19982 JSR .1
19988 LDS .1
19994 STS .1
20000 SUBA 2
20006 CMPA 2
20012 SBCA 2
20018 SUBD 2
20024 ANDA 2
20030 BITA 2
20036 LDAA 2
20042 STAA 2
20048 EORA 2
20054 ADCA 2
20060 ORAA 2
20066 ADDA 2
20072 CPX  2
20078 JSR  2
20084 LDS  2
20090 STS  2
20096 SUBB#1
20102 CMPB#1
20108 SBCB#1
20114 ADDD#2
20120 ANDB#1
20126 BITB#1
20132 LDAB#1
20138
20144 EORB#1
20150 ADCB#1
20156 ORAB#1
20162 ADDB#1
20168 LDAD#2
20174
20180 LDX #2
20186
20192 SUBB 1
20198 CMPB 1
20204 SBCB 1
20210 ADDD 1
20216 ANDB 1
20222 BITB 1
20228 LDAB 1
20234 STAB 1
20240 EORB 1
20246 ADCB 1
20252 ORAB 1
20258 ADDB 1
20264 LDAD 1
20270 STAD 1
20276 LDX  1
20282 STX  1
20288 SUBB.1
20294 CMPB.1
20300 SBCB.1
20306 ADDD.1
20312 ANDB.1
20318 BITB.1
20324 LDAB.1
20330 STAB.1
20336 EORB.1
20342 ADCB.1
20348 ORAB.1
20354 ADDB.1
20360 LDAD.1
20366 STAD.1
20372 LDX .1
20378 STX .1
20384 SUBB 2
20390 CMPB 2
20396 SBCB 2
20402 ADDD 2
20408 ANDB 2
20414 BITB 2
20420 LDAB 2
20426 STAB 2
20432 EORB 2
20438 ADCB 2
20444 ORAB 2
20450 ADDB 2
20456 LDAD 2
20462 STAD 2
20468 LDX  2
20474 STX  2
```

LISTING 2
****

```
1000 REM*PROGRAM TO LOAD MNEMONICS OF OP
CODES INTO MEMORY.
1010 REM DATA INTO MEMORY*
1015 CLEAR 100,18944
1020 INPUT"HOW MANY OPCODES ";NOPC
1025 NBYT=NOPC*6
1030 INPUT"WHAT IS FIRST MEMORY LOCATION
";SMMRY
1050 FOR N=SMMRY TO (SMMRY+NBYT-1)   STE
P 6
1055 PRINT"NEXT MNEMONIC GOES INTO ";N
1060 PRINT"WHAT IS MNEMONIC STRING"
1065 INPUT MNEM$
1067 IF MNEM$=""THEN MNEM$="        "
1070 FOR J=0 TO 5
1080 MNEM=ASC(MNEM$)
1090 POKE N+J,MNEM
1100 MNEM$=RIGHT$(MNEM$,5-J)
1110 NEXT J
1120 NEXT N
1130 END
2000 REM*PROGRAM TO SAVE A BINARY FILE O
R MACHINE CODE
2005 REM*SIMULATES CSAVEM
2007 REM*FORMAT FOR CSAVEM IS-
2010 REM FIRST,LAST,START,"FILENAME"
2020 CLS:A$="12345678901234567890123456
78901234567890123456789012345678901234
567890123456789012345678901234567890123"
2030 B=VARPTR(A$):IFPEEK(B)<>53THENSTOP
2040 X=256*PEEK(B+2)+PEEK(B+3)
2050 FOR Z=X TO X+52
2060 READ Y:W=W+Y:PRINT Z,Y;W
2070 POKE Z,Y:NEXT
2080 IFW<>8373 THENPRINT"DATA ERROR":STO
P
2090 EXEC X:END
2100 DATA 206,66,178,223,244,189
2110 DATA 231,255,189,239,76,255
2120 DATA 66,108,189,234,47,189
2130 DATA 239,76,8,255,66,113
2140 DATA 189,234,47,189,239,76
2150 DATA 255,66,106,189,234,47
2160 DATA 134,2,183,66,103,189
2170 DATA 252,139,254,66,108
2180 DATA 189,252,96,189,226,113
3000 REM*PROGRAM TO CHECK OPCODES
3002 PRINT"MNEMONICS ARE FOUND   FROM 18
944 TO 20474"
3005 INPUT"ADDRESS OF FIRST MNEMONIC ";F
ADDR
3006 INPUT"ADDRESS OF LAST MNEMONIC ";LA
DDR
3010 FOR N=FADDR TO LADDR STEP6
3020 LPRINTN;
3030 FOR M=1 TO 6
3040 LPRINTCHR$(PEEK(M+N-1));
3050 NEXT M
3060 LPRINT""
3070 NEXT N
3080 END
```

LISTING 3
****

```
1000 REM*BRUTE FORCE DISASSEMBLER
                    DECEMBER, 1984.
```

BRUTE FORCE

```
1020 INPUT"WHAT IS START ADDRESS";SADDRS
1030 INPUT"HOW MANY BYTES TO DISASSEMBLE
";NBYT
1040 INPUT"WHAT IS TERMINATING MNEMONIC"
;TMNEM$
1045 TMNEM$=LEFT$(TMNEM$+"    ",5)
1050 N=SADDRS
1051 REM*START OF LOOP*
1060 OP=PEEK(N)
1070 PNTR=OP*6+18944
1080 P0=PNTR:P1=PNTR+1:P2=PNTR+2
1090 P3=PNTR+3:P4=PNTR+4:P5=PNTR+5
1110 MN$=CHR$(PEEK(P0))+CHR$(PEEK(P1))+C
HR$(PEEK(P2))+CHR$(PEEK(P3))+CHR$(PEEK(P
4))
1120 D=PEEK(P5)-48
1125 IF D=-16 THEN PRINT"INVALID OPCODE"
:END
1130 ON D+1 GOTO 1155,1150,1140
1140 LSB=PEEK(N+2)
1142 MMCNT=LSB:NMDG=2:GOSUB3000
1143 LSB$=HEX$
1150 MSB=PEEK(N+1)
1152 MMCNT=MSB:NMDG=2:GOSUB3000
1153 MSB$=HEX$
1155 MMCNT=N:NMDG=4:GOSUB3000
1156 ADDR$=HEX$
1157 LPRINT N;ADDR$;
1160 LPRINT TAB(16) MN$;
1165 GOSUB 4000 REM*CALC,BRANCH
1170 ON D+1 GOSUB 1220,1230,1240
1180 IF MN$=TMNEM$ THEN END
1190 N=N+D+1
1200 IF N<SADDRS+NBYT-1 THEN1051
1210 END
1220 LPRINT"":RETURN
1230 LPRINTMSB$;DST$;
1231 IF LEFT$(MN$,1)="B"THENLPRINTDST;
1232 LPRINT"":RETURN
1240 LPRINTMSB$;LSB$:RETURN
2000 REM*CONVERT DEC.DIG. TO HEX DIG.
2010 IF DEC<10 THEN 2040
2020 HXDG$=CHR$(55+DEC)
2030 GOTO 2050
2040 HXDG$=RIGHT$(STR$(DEC),1)
2050 RETURN
3000 REM*CONVERT DEC.INTO HEX
3010 HEX$=""
3020 FOR H=1 TO NMDG
3030 QUOT=MMCNT/16
3040 MMCNT=INT(QUOT)
3050 RMNDR=16*(QUOT-MMCNT)
3060 DEC=RMNDR:GOSUB2000
3070 HEX$=HXDG$+HEX$
3080 NEXT H
3090 RETURN
4000 REM*CALC. BRANCH DESTIN.
4010 IFLEFT$(MN$,1)<>"B"THENDST$="":GOTO
 4060
4020 DST=N+D+1+MSB
4030 IFMSB>127THENDST=DST-256
4040 MMCNT=DST:NMDG=4:GOSUB3000
4050 DST$="->"+HEX$+"/"
4060 RETURN
```

LISTING 4

```
63278 F72E    LDAA#FF
63280 F730    STAA 00
63282 F732    LDAA#01
63284 F734    STAA 01
63286 F736    LDAA#01
63288 F738    STAA 03
63290 F73A    LDAA EA
63292 F73C    CMPA#55
63294 F73E    BNE  0A->F74A/ 63306
63296 F740    LDX  4221
63299 F743    LDAA,00
63301 F745    DECA
63302 F746    BNE  02->F74A/ 63306
63304 F748    JMP  ,00
63306 F74A    LDX #0080
63309 F74D    CLR  ,00
63311 F74F    INX
63312 F750    CPX #0100
63315 F753    BNE  F8->F74D/ 63309
63317 F755    LDX #41FD
63320 F758    INX
63321 F759    LDAA,02
63323 F75B    COM  ,02
63325 F75D    LDAB,02
63327 F75F    CLR  ,02
63329 F761    COMA
63330 F762    CBA
63331 F763    BEQ  F3->F758/ 63320
63333 F765    STX  4250
63336 F768    STX  A1
63338 F76A    STX  9D
63340 F76C    LDAD 9D
63342 F76E    SUBD#0064
63345 F771    STAD 9B
63347 F773    LDS  9B
63349 F775    LDX #F7CF
63352 F778    LDAD#00EB
63355 F77B    BSR  30->F7AD/ 63405
63357 F77D    LDX #F7DE
63360 F780    LDAD#4200
63363 F783    BSR  28->F7AD/ 63405
63365 F785    LDAA#39
63367 F787    LDX #4285
63370 F78A    STAA,00
63372 F78C    INX
63373 F78D    CPX #42AF
63376 F790    BNE  F8->F78A/ 63370
63378 F792    COM  42AF
63381 F795    LDX #4346
63384 F798    STX  93
63386 F79A    JSR  E3CF
63389 F79D    JSR  FBD4
63392 F7A0    LDX #F80F
63395 F7A3    JSR  E7AB
63398 F7A6    LDAA#55
63400 F7A8    STAA EA
63402 F7AA    JMP  E271
63405 F7AD    STAD BF
63407 F7AF    LDAB,00
63409 F7B1    INX
63410 F7B2    LDAA,00
63412 F7B4    STX  C1
63414 F7B6    LDX  BF
63416 F7B8    STAA,00
63418 F7BA    INX
63419 F7BB    STX  BF
63421 F7BD    LDX  C1
63423 F7BF    DECB
63424 F7C0    BNE  EF->F7B1/ 63409
63426 F7C2    RTS
```

BRIAN McLAUGHLIN

Professional programmers love to dress up their creations with superb title pages. While it is sometimes downright painful to have to wade through screeds and screeds of fancy title pages a well presented "Cover" can improve the quality and appearance of your program no end.

Brian McLaughlin is one such chap who takes the trouble to add the finishing touches to his programs and presented here is a program he uses to produce the title screens for his work.

Enter your own heading into the DATA statements in line 567. You may also have to alter the READ statements if you get an OD (Out of Data) ERROR. Play around with this one or develop your own style using some of these ideas. It's well worth the effort to present your work well.

The Listing:

```
0 REM***MICO-MAGIC***BY BRIAN MC
LAUGHLIN***
1 REM***FOR LARGE LETTER HEADING
S***
2 REM***TRY EXPERIMENTING AT THE
 BEGINNING OF YOUR PROGRAMS***
10 CLS0:GOTO100
100 CLS0:CLEAR40:GOSUB1000
110 CLEAR100:CLS
130 DIMAZ$(26):QP=128:GOSUB500
140 CLS0:PRINT@34,"_try_a_few_ex
periments_";:FORI=1TO2000:NEXTI:
CLS0:PRINT@34,"_bye_bye_";:FORI=
1TO2000:NEXTI:CLS:GOTO100
500 AZ$(1)="0103050103050715O7":
AZ$(2)="0103130103050303l5":AZ$(
3)="0103070515150303O7":AZ$(4)="
0103130515050303l5":AZ$(5)="0103
070701071503030?"
510 AZ$(6)="0103070107150715l5":
AZ$(7)="0103070511050303O7":AZ$(
8)="0515050103050715O7":AZ$(9)="
0301071505150303O7":AZ$(10)="151
505131505110315"
520 AZ$(11)="0509150113150711l5"
:AZ$(12)="0515150515150303l5":AZ
$(13)="0414050507050715O7":AZ$(1
4)="0415050506050715O7":AZ$(15)=
"0103050515050303O7"
530 AZ$(16)="0103050103070715l5"
:AZ$(17)="0103050506070307O7":AZ
$(18)="0103050101070711l5":AZ$(1
9)="0103070303050303O7":AZ$(20)=
"0301071505151507l5"
540 AZ$(21)="0515050515050303O7"
:AZ$(22)="0515051010151507l5":AZ
$(23)="0515050406050715O7":AZ$(2
4)="0614071406150715O7":AZ$(25)=
"0614071505151507l5"
550 AZ$(26)="0302071407150303O7"
551 DATA M,I,C,O,M,A,G,I,C
552 QZ=394:FORI=1TO4:READQZ$:GOS
UB570:SOUNDI*10+50,1:NEXT:FOR I=
1TO5:PRINT@480,"":SOUNDI*10+50,1
:NEXT
553 QZ=392:FORI=1TO5:READQZ$:GOS
UB570:SOUNDI*10+150,1:NEXT:FORI=
1TO4:PRINT@480,"":SOUNDI*5+150,1
:NEXT:FORI=1TO1000:NEXT
561 CLS:QZ=388:FORI=1TO8:READQZ$
:GOSUB570:SOUNDI*5+160,1:NEXT:FO
RI=1TO3:PRINT@480,"":SOUNDI*5+16
0,1:NEXT:FORI=1TO2000:NEXT
565 QZ=397:FORI=1TO2:READQZ$:GOS
UB570:SOUNDI*5+160,1:NEXT:FORI=1
TO3:PRINT@480,"":SOUNDI*5+160,1:
NEXT:FORI=1TO2000:NEXT
```

```
566 QZ=394:FORI=1TO4:READQZ$:GOS
UB570:SOUNDI*5+160,1:NEXT:FORI=1
TO3:PRINT@480,"":SOUNDI*5+160,1:
NEXT:FORI=1TO2000:NEXT
567 DATAH,E,A,D,I,N,G,S,G,O,H,E,
R,E:GOTO140
570 QL=LEN(QZ$)
580 FOR QY=1 TO QL:QA= ASC( MID$
(QZ$,QY,1))
590 IF QA<65 OR QA>90 THEN 660
600 QA=QA-64
610 FOR QI=0 TO 2:FOR QJ=1 TO 5
STEP 2
620 QK=QJ+QI*6
630 QX= VAL( MID$(AZ$(QA),QK,2))
640 PRINT@QZ, CHR$(QX+QP);:QZ=QZ
+1
650 NEXT QJ:QZ=QZ+29:NEXT QI:QZ=
QZ-93
660 NEXTQY:RETURN
1000 REM
1005 A=RND(14)*32:B=RND(15)
1007 Z$=CHR$(143+RND(7)*16)
1008 PRINT@A+B,Z$;:PRINT@A+31-B,
Z$;:PRINT@448-A+31-B,Z$;
1010 CLS0:B$=CHR$(128):W$=CHR$(2
07)
1011 PRINT@10,W$+W$+W$+W$+W$+B$+
W$;
1012 PRINT@42,W$+B$+W$+B$+W$;:PR
INT@74,W$+B$+W$+B$+W$+B$+W$+B$+W
$+CHR$(156)+B$+W$+CHR$(156)+W$;
1015 PRINT@106,W$+B$+W$+B$+W$+B$
+W$+B$+W$+CHR$(147)+B$+W$+CHR$(1
47)+W$;
1020 PRINT@169,W$+W$+W$+W$+W$+B$
+B$+B$+B$+B$+B$+B$+B$+B$+W$;:PRI
NT@201,W$+B$+W$+B$+W$;
1025 PRINT@233,W$+B$+W$+B$+W$+B$
+W$+CHR$(156)+CHR$(154)+B$+W$+CH
R$(156)+W$+B$+W$+B$+W$+CHR$(156)
;
1030 PRINT@265,W$+B$+W$+B$+W$+B$
+W$+CHR$(147)+CHR$(155)+B$+W$+CH
R$(147)+W$+B$+W$+B$+W$+CHR$(147)
;
1035 PRINT@309,W$;:PRINT@339,CHR
$(156)+CHR$(156)+CHR$(156);
1040 A=16:FORX=1TO6:C=143+(16*X)
:FORY=4TO30:S=RND(235):SOUNDS,1:
PRINT@127+(X*32)+Y,CHR$(C);:NEXT
Y,X
1050 K1$="   MICO-MAGIC   ":K$=K1
$:GOSUB2000
1065 K4$="BY BRIAN MCLAUGHLIN":K
$=K4$:GOSUB2000:GOTO2002
2000 PRINT@231,"
   ";
2001 FORI=1TO19:S=RND(235):SOUND
S,1:L$=LEFT$(K$,I):PRINT@250-I,L
$;:FORD=1TO10:NEXTD,I:FORD=1TO25
0:NEXTD
2002 RETURN
```

Music

Brian Mclaughlin

MiCo MUSIC is designed to enable you to develop DATA strings of music which can then be played by the computer.

Enter your own DATA from lines 2000 onwards. Leave line 10000 as it is. It signals the end of the music. The instructions run from line 50 through to line 170. Lines 175 to 930 read the data and play the music. Lines 20000 onwards are a subroutine to present the program title page.

Greensleeves is used here to demonstrate the workings of MiCo Music. Have Fun and perhaps even some Merry Melody Making.

The Listing:

```
1 CLS0:CLEAR50
2 A=RND(14)*32:B=RND(15):Z$=CHR$
(143+RND(7)*16)
3 PRINT@A+B,Z$;:PRINT@A+31-B,Z$;
:PRINT@448-A+B,Z$;:PRINT448-A+31
-B,Z$;:P=P+1:IFP=50THEN10
10 CLS0:B$=CHR$(128):W$=CHR$(207
):PRINT@10,W$+W$+W$+W$+W$+B$+W$;
13 PRINT@42,W$+B$+W$+B$+W$;:PRIN
T@74,W$+B$+W$+B$+W$+B$+W$+B$+W$+
CHR$(156)+B$+W$+CHR$(156)+W$;
14 PRINT@106,W$+B$+W$+B$+W$+B$+W
$+B$+W$+CHR$(147)+B$+W$+CHR$(147
)+W$;:PRINT@169,W$+W$+W$+W$+W$+B
$+B$+B$+B$+B$+B$+B$+B$+B$+CHR$(147)
;
16 PRINT@201,W$+B$+W$+B$+W$+B$+C
HR$(147)+B$+CHR$(147)+B$+CHR$(14
7)+CHR$(147)+B$+CHR$(147)+B$+CHR
$(147)+CHR$(147);
17 PRINT@233,W$+B$+W$+B$+W$+B$+W
$+B$+W$+B$+W$+CHR$(147)+B$+W$+B$
+W$;
18 PRINT@265,W$+B$+W$+B$+W$+B$+W
$+CHR$(147)+W$+B$+CHR$(147)+W$+B
$+W$+B$+W$+CHR$(147);
19 GOSUB20000
49 REM***PROGRAM STARTS HERE***
50 CLS:PRINT@10,"MICOMUSIC"
```

```
51 PRINT@34,"COMPARRISON SCALE O
F VALUES":PRINT@131,"MUSIC SCALE
 WITH SHARPS(*)":PRINT@195,"0CCD
DEFFGGAABCCDDEFFGGAABC"
55 PRINT@227,"@ABCDEFGHIJKLMNOPQ
RSTUVWXY":PRINT@165,"* *  * * *
 * *  * * *"
70 PRINT@265,"FROM MIDDLE C(A) TO
 C(Y) ABOVE":PRINT@106,"THE STAV
E":PRINT@291,"MUSIC SCALE WITH F
LATS (0)"
77 PRINT@325,"0 0  0 0 0  0 0  0
 0 0":PRINT@387,"@ABCDEFGHIJKLMN
OPQRSTUVWXY":PRINT@355,"OCDDEEFG
GAABBCDDEEFGGAABBC"
110 PRINT@449,"PRESS D FOR DURAT
ION VALUES"
111 D$=INKEY$:IFD$=""THEN111
112 IFD$="D"THEN114
114 CLS:PRINT@38,"DURATION VALUE
S":PRINT@97,"SEMIQUAVER=2 PLUS A
 DOT=3":PRINT@133,"QUAVER=4 PLUS
 A DOT=6":PRINT@163,"CROTCHET=8
PLUS A DOT=<"
118 PRINT@198,"MINUM=@ PLUS A DO
T =H":PRINT@226,"SEMIBREVE=P":PR
INT@296,"REST VALUES":PRINT@321,
"ARE THE EQUIVALENT OF DURATION"
122 PRINT@353,"BUT USE @ FOR ZER
O TONE":PRINT@417,"PRESS KEY P I
F UNDESTOOD"
123 PRINT@417,"PRESS KEY P IF UN
DERSTOOD"
139 R$=INKEY$:IFR$=""THEN139
140 IFR$="P"THEN150
145 IFR$<>"P"THEN139
150 CLS
160 PRINT@162,"INSERT YOUR OWN S
ONG DATA AT":PRINT@194,"2000 ONW
ARDS USING THE SCALE":PRINT@226,
"OF COMPARRISON, PRESS P NOW"
163 PRINT@258,"AND CONTINUE TRAN
SPOSITION"
170 PRINT:PRINT"PRESS (P) AND HE
AR THE SONG (GREENSLEEVES)"
175 R$=INKEY$:IFR$=""THEN175
176 IFR$="P"THEN180
180 CLEAR50:DIM N(26)
190 RESTORE
515 A=ASC("A")-2
530 FORJ=1TO26:READN(J):NEXTJ
560 READ V$
570 IFV$="#"THEN 190
590 READW$:P=1:T=0:C=0:IFC=TTHEN
630
630 Y=ASC(MID$(W$,P,1))
631 Y=Y-48:IFY=0THEN560
650 X=ASC(MID$(V$,P,1))
660 P=P+1:T=T+Y:IFN(X-A)=0THEN93
0
680 SOUNDN(X-A),Y:C=C+1:GOTO630
930 FORK=0TOY*8:NEXTK
1000 DATA0,131,138,145,151,156,1
62,167,172
1010 DATA177,181,185,189,193,196
,200,203
1020 DATA206,209,211,214,216,218
,220,222,234
2000 DATA"JMOQRQOLHJLMJJIJLI","4
8462484624844248 40"
2002 DATA"EEMOQRQOLHJLMLJIGI","8
4846248462462462 40"
2004 DATA"JJJTTSQOLHJLMJJIJ","84
<<822846248462 40"
2006 DATA"LIEETTSQOLHJLMJIGIJ","
84<4684284624844248H0"
10000 DATA"#"
20000 A=16:FORX=1TO4:C=143+(16*X
):FORY=5TO29:S=RND(230):SOUNDS,1
:PRINT@127+(X*32)+Y,CHR$(C);:NEX
TY,X
20100 K1$="    MICO-MUSIC ":K$=K
1$:GOSUB20500:K2$="BY BRIAN MCLA
UGHLIN":K$=K2$:GOSUB20500:FORI=1
TO500:NEXTI:GOTO49
20500 PRINT@231,"
    ";:FORI=1TO19:L$=LEFT$(K$,I)
:PRINT@250-I,L$;:S=RND(235):SOUN
DS,1
20510 FORD=1TO5:NEXTD,I:FORD=1TO
150:NEXTD:RETURN
```

Grahame Pollock

MORE

I would like to say that it is great to see those programs rolling in. The more programs we see in MiCo, the better off we'll be. After all, the very best way to learn computing is to key in a program and debug it. The bugs are mostly our own typing errors or they might be those sneaky little bugs that Greg used to put in just to see if we were all still thinking. For example he slipped a bug into line 310 of CONTENTS in MiCoOz#5 and MiCoOz#7. There were a couple of THEN's which should have been +'s (both easy to fix using LITTLE-E).

Greg also told us that GUESS and POKER MACHINE (Jan MiCo) were 20K. In fact they will fit into 4K without any trouble. It's surprising what will fit into a 4K machine. If a program is made on a 20K MiCo or on a 16K E.C.B. CoCo a person may think that it won't fit into 4K.

Usually there are two things about a program which will determine whether or not it will fit into 4K. Firstly the program itself may be very long, containing too many numbers or letters and spaces to fit into the 4K memory and secondly it may contain DIMensioned space that's too big.

TOWERS of Hanoi (Jan) and SONGTIME (Feb) are examples of the first type (long) and ANIMALS (Jan) is an example of the second type (big DIM).

If you try to CLOAD one of the long programs from a tape like MiCoOz then you'll get an OM ERROR when the memory limit is reached. You might just be able to type in some of these long programs into your 4K MiCo if you're careful AND if you have a listing for it. If you want to try to fit it in you've really got to cut out ANYTHING that's unnecessary. You'll usually find some of these lines right at the beginning of the program. Look at the following program!

```
10 REM PROGRAM
20 REM BY G. POLLOCK
30 REM FOR THE 20K MC-10
40 CLS0
50 PRINT@ 200, "WELCOME";
60 PRINT@ 234, "TO";
70 PRINT@, 264, "PROGRAM";
80 FOR R=1 TO 8
90 CLSR
100 SOUND20*8,RND(10)
110 NEXT R
120 PRINT "DO YOU WISH TO CONTIN
UE WITH THIS PROGRAM";
130 INPUT"IF SO, PRESS ENTER";P$
140 ON RND(2) GOTO 40,80
```

This program could become:

```
40 CLS0:PRINT@200,"WELCOME";:PR
INT@234,"TO";:PRINT@264,"PROGRAM
";
80 FORR=1TO8:CLSR:SOUND20*R,RND(1
0):NEXT:INPUT"PRESS ENTER TO CONT
INUE";P$:ONRND(2)GOTO40,80
```

Notice that most spaces have been eliminated and that the number of lines has been reduced. Notice also that lines 40 and 80 still exist. This is important because the program actually refers to these lines.

The second type of program is one with big DIMensions as I mentioned before. This type of program will CLOAD but will not RUN. You'll usually find these big DIM's at the beginning of the program. This type might work if you chose the largest DIM which will allow the program to RUN in the memory space which is available to you. For example, you might have to change

```
10 DIM A$(200)
```

to

```
10 DIM A$(150)
```

or

```
10 DIM A$(100)
```

so that the program will RUN.

While I'm on the subject of OM ERROR's, there is another type that I've encountered recently. I wanted to try out MC10CONV from MiCoOz#1 so I went up to Tandy (Minto Mall). The staff were extremely helpful. They let me use their demo CoCo to convert a program to MiCo'ese. I followed the instructions but I got an immediate OM ERROR when CLOADing into the MC-10. Even a very short (single line) program typed into the CoCo and converted with MICOCONV gave an immediate OM ERROR when CLOADed into the MC-10. It seems as if the program is being CLOADed into the wrong spot in memory. Maybe the program works for the 20K MiCo. If you've had any luck with MICOCONV, perhaps you could let me know.

I'm sending in three short programs this month, POKETEST, REPEAT and ROLLOVER.

I wrote POKETEST because I wanted to find out more about POKEs and PEEKs but, to put it simply PEEK looks at a memory location and POKE puts a number of your choice into a specific memory location. The things that the computer does will depend on which number is in what memory location, and sometimes the number on the memory location will depend on the things that you do at the keyboard. Here are some things for you to try:

```
PEEK(21000)
```

This will only give you an SN ERROR because you've told the computer to look

at a memory location but you haven't told it to show you that number.

Now try: PRINTPEEK(21000)

then: POKE21000,120

then: PRINTPEEK(21000)

The main thing that you will notice is that a POKE puts a number in and a PEEK lets you look at a number that is stored in the computer.

In POKETEST you will see the POKES that give you screen print positions.

In repeat you will see how PEEKs can be used to let you hold down a key and have it printed on the screen for as long as it is held down. A similar type of thing is seen in ROLLOVER but pokes must be used as well in this section of the memory.

Try this short program:

```
10 CLS
20 PRINT@0,PEEK(49150)
30 GOTO10
```

It will give you a continuous display of the number which is stored in memory location 49150. Now try pressing down and holding a few keys. You might notice that only the keys on the bottom row of the ROLLOVER TABLE will work (pg44 Feb MiCo).

I've found another memory location which might be of some interest to you.

try
```
PRINTPEEK(16925)
```

and then try
```
PRINTPEEK(16926)
```

You should get 4 and 94. These numbers in these memory locations control the delay (debounce) time of the computer keyboard. You can change this delay time by POKEing into these memory locations

try
```
POKE16925,200
```

Now press any key and hold it down. Try to type your name!!! See how slowly the keyboard responds. See if you can speed up the keyboard response.

Just one more thing before I go. I just want to let you know that <shift> @ will stop the MC-10 immediately no matter what it is doing. Most of you will already know this but some may not. It took me quite some time to find out. I knew that CONTROL S did the same thing on the APPLE and I tried the control key with all the other keys but I didn't try <shift> @ while it was listing. Now press any other key to start it again. With practice you should be able to LIST to any point by keeping your right thumb on the SHIFT key and alternating your index finger between the ENTER key and @ key. The BREAK key will return to the OK prompt with the listing halted at that point.

Listing 1:

```
5 G=25
10 FORX=16384 TO 20479
20 FORY=0TO255
25 CLS:PRINT@200,;
30 PRINT"POKE";X;",";Y;
40 POKEX,Y
45 I$=INKEY$:IFI$="S"THENG=G/5
46 IFI$="A"THENG=G*5
50 FORT=1TOG:NEXTT
60 NEXTY:NEXTX
```

Listing 2:

```
4 REM REPEAT BY G.POLLOCK
5 P=49150
6 PRINT"REPEAT"
7 PRINT"TRY PRESSING AND HOLDING
 DOWN THE KEYS
8 PRINT"/,W,O,7,G,space"
10 IFPEEK(P)=223THENPRINT"/";:SO
UND200,1
20 IFPEEK(P)=251THENPRINT"W";:SO
UND200,1
30 IFPEEK(P)=253THENPRINT"O";:SO
UND200,1
40 IFPEEK(P)=247THENPRINT" ";:SO
UND200,1
50 IFPEEK(P)=254THENPRINT"G";:SO
UND200,1
60 IFPEEK(P)=239THENPRINT"7";:SO
UND200,1
70 GOTO10
```

Listing 3:

```
4 REM ROLLOVER BY G.POLLOCK
5 A$=INKEY$:IFA$=""THEN5
8 FORX=16945TO16952
10 IFPEEK(X)<>255THENPRINTA$;:SO
UNDASC(A$),1:POKEX,255:GOTO5
25 NEXT
30 GOTO5
```

MiCo TELEPHONE DIRECTORY

Brian McLaughlin

One of the major uses for computers is data storage and retrieval. That data can be anything from Cooking Recipes to Phone Numbers to Bank Account Details. You name it, there is probably someone storing that data using a computer.

With the MiCo, TANDY were a little unfair to us in that they provided no easy way to store Data Files. Notice I used the word easy, not impossible. One of the easiest ways we have of storing data is to include our data in the main body of the program as DATA statements. In fact using this system in many cases (especially for storing phone numbers) is better than having to maintain a separate set of data files. Everything is loaded in and ready to run once the BASIC program is loaded.

Type in the listing putting your own DATA in lines between 900 and 1020. Line 1020 flags the end of the DATA File so do not alter this line. The format of the DATA statement is firstly the name you will be entering as your search field, followed by a comma and the details (address and phone number) of that person.

When you run the program it will ask for the name you wish to find, search for that name and print out the address and phone number. If you want the record to be printed to the printer change line 330 to read:

```
330 LPRINT Q$, T$(M):LPRINT CHR$(13)
```

I hope that you find this a useful utility. It does not have the sophistication (complexity) of a CoCo disk system but who needs all those problems.

The Listing:

```
1 REM***MICO TELEPHONE DIRECTORY
***BY BRIAN MCLAUGHLIN***
2 REM***INSERT YOUR OWN TELEPHON
E NUMBERS AND ADDRESSES BEGINNIN
G AT LINE 910***
3 REM***NO PRINTER***"
4 REM***REMOVE REM STATEMENTS 1
DECEMBER, 1984.
TO 4 BEFORE RUNNING PROGRAM***
25 CLEAR100:GOSUB2000
30 CLS:DIMN$(20):DIMT$(20):I=1
50 READN$(I),T$(I)
60 IFN$(I)="ZZZZ"THEN100
70 I=I+1:GOTO50
100 N=I
150 CLS:INPUT"NAME PLEASE";Q$
220 L=1:H=N
230 IFH-L=1THEN500
240 M=INT((L+H)/2)
260 IFQ$=N$(M)THEN320
270 IFQ$<N$(M)THEN300
280 L=M:GOTO230
300 H=M:GOTO230
320 REM***END OF SEARCH***
330 CLS:PRINT@32:PRINT Q$, T$(M)
;
335 FORS=110TO230STEP10:SOUNDS,1
:NEXTS
346 FORI=1TO3000:NEXTI:CLS:GOTO6
00
500 PRINTQ$;" IS NOT IN DIRECTOR
Y "
505 FORS=100TO50STEP-5:SOUNDS,1:
NEXTS:FORI=1TO2000:NEXTI
600 CLS:INPUT"DO YOU WISH FOR AN
OTHER NUMBER";R$
610 IFR$="YES"THEN150
611 IFR$="NO"THEN612
612 CLS:PRINT@34,"PROGRAM WILL N
OW NEW AND CLOAD";:PRINT@66,"THE
 NEXT PROGRAM ON TAPE";:GOTO614
614 PRINT:INPUT"PRESS ENTER TO C
ONTINUE";X$:PRINT@228,"NOW DOING
 IT, PRESS PLAY ON RECORDER";:EX
EC64874
620 END
900 DATAAAAAA,0000
910 REMDATA HERE
911 DATAAUSTRALIAN RAINBOW,075-5
10015
915 DATABRIAN MCLAUGHLIN,P.O.BOX
-7 COOMA NSW 2630-50
919 DATAGRAHAM MORPHETT,075-5100
15
1020 DATAZZZZ,9999
2000 CLS0:A=RND(14):B=RND(15):Z$
=CHR$(143+RND(7)*16)
2001 PRINT@A+B,Z$;:PRINT@A+31-B,
Z$;:PRINT@448-A+B,Z$;:PRINT@448-
A+31-B,Z$;
2002 P=P+1:IFP=50THEN2003
2003 CLS0:B$=CHR$(128):W$=CHR$(2
07)
2006 PRINT@10,W$+W$+W$+W$+W$+B$+
W$;
2007 PRINT@42,W$+B$+W$+B$+W$;
2008 PRINT@74,W$+B$+W$+B$+W$+B$+
W$+B$+W$+CHR$(156)+B$+W$+CHR$(15
6)+W$:
```

```
2009 PRINT@106,W$+B$+W$+B$+W$+B$
+W$+B$+W$+CHR$(147)+B$+W$+CHR$(1
47)+W$;
2010 PRINT@364,"_telephone_";:PR
INT@396,"_directory_";
2020 FORI=1TO500:NEXTI
2050 A=16:FORX=1TO5:C=143+(16*X)
:FORY=5TO29:S=RND(235):SOUNDS,1:
PRINT@127+(X*32)+Y,CHR$(C);:NEXT
Y,X
2055 K1$="TELEPHONE DIRECTORY":K
$=K1$:GOSUB2090:K2$="BY BRIAN MC
LAUGHLIN":K$=K2$:GOSUB2090
2056 K3$="  INSERT YOUR OWN ":K$
=K3$:GOSUB2090:K4$=" TELEPHONE N
UMBERS ":K$=K4$:GOSUB2090
2057 K5$="  AND ADDRESSES IN ":K
$=K5$:GOSUB2090:K6$=" DATA STATE
MENTS ":K$=K6$:GOSUB2090
2060 K7$="AND THE NUMBER OF ":K$
=K7$:GOSUB2090:K8$="PERSON YOU R
EQUIRE ":K$=K8$:GOSUB2090
2065 K9$="WILL BE DISPLAYED ":K$
=K9$:GOSUB2090:GOTO3000
2090 PRINT@231,"
   ";:FORI=1TO19:L$=LEFT$(K$,I):
S=RND(235):SOUNDS,1:PRINT@250-I,
L$;
2091 FORD=1TO10:NEXTD,I:FORD=1TO
200:NEXTD:RETURN
3000 RETURN
```

I would like to be included as a MiCo contact. I am an hour's drive north of the Central Coast at Budgewoi, and there may be some who are reluctant to go that far to meet but we already have 4 people with CoCo's in the area so we have a start!

Hope to meet you at CoCoConf,

John Wallace.
Budgewoi. NSW.

John,
We depend a little upon the authors to explain their own work. Where possible we will try to assist with fuller descriptions of program functions. You will like the Basic Tutorial section that we have introduced. I'm hoping that Wim de Pruitt (sorry if I got the spelling wrong mate!) in Tassie will write for that column. He wrote me a letter disclaiming his skills, and in the process proving that what I said of him was true - he knows how to string a few words together!

Hope to hear that your group is achieving good things in the not too distant future.

Graham.

SPACE BAR BANDIT

by Tom Lehane

4K MC10 (16K CoCo)

I have seen so many Poker Machine programs where the emphasis seems to be placed on superb colour graphics. Yet is it not true that the simpler a program is the more elegant it becomes (providing the purpose of the program is not sacrificed).

SPACE BAR BANDIT was written to suit a 4K MiCo or non-extended CoCo. The graphics are deliberately simple and yet after intensive research it is clearly apparant that the program does all that is required of a poker machine simulation.

A record of your winnings (losses) is maintained constantly on the screen. Payouts are described in the instructions and as the name suggests, all that is required for you to liquidate your capital is to simply key repeatedly the Space Bar.

The Listing:

```
10 CLS
20 PRINT@424,"SPACE BAR BANDIT"
30 FOR X=1 TO 12
40 PRINT@511,""
50 FOR TL=1 TO 100:NEXT TL
60 NEXT
70 PRINT@356, CHR$(130)+ CHR$(12
9);
80 PRINT@389, CHR$(138)+ CHR$(13
2);
90 PRINT@392,"BY TOM LEHANE"
100 PRINT@104,"JACKPOT PAYS $50.
0"
110 PRINT@167,"THREE OF A KIND $
10"
120 PRINT@232,"TWO OF KIND $2.0
130 PRINT@480,"PRESS ENTER";:INP
UTR
140 GOTO  360
150 C=C+1
```

```
160 PRINT@45,C;
170 FOR S=1 TO 25
180 T= RND(5)+34
190 O= RND(5)+34
200 M= RND(5)+34
210 PRINT@269, CHR$(T);
220 PRINT@271, CHR$(O);
230 PRINT@273, CHR$(M);
240 NEXT
250 PRINT@172,""
260 IF T=36 AND O=36 AND M=36 TH
EN 300
270 IF T=O AND O=M THEN 320
280 IF T=O AND M<> 40 THEN 340
290 GOTO  680
300 PRINT@424,"*** JACKPOT ***";
310 GOTO  890
320 PRINT@424,"THREE OF A KIND";
330 GOTO  830
340 PRINT@424,".TWO OF A KIND.";
350 GOTO  860
360 CLS 0
370 PRINT@98,"WINNINGS >$"
380 PRINT@162,"PAY OUT..>"
390 PRINT@34,"MONEY IN >$"
400 PRINT@269,"$";:PRINT@271,"$"
;:PRINT@273,"$";
410 FOR X=24 TO 37
420 SET(X,14,8)
430 SET(X,19,8)
440 NEXT X
450 FOR Y= 14 TO 19
460 SET(24,Y,8)
470 SET(37,Y,8)
480 NEXT Y
490 FOR X=22 TO 39
500 SET(X,20,4)
510 SET(X,23,4)
520 NEXT X
530 FOR Y=20 TO 23
540 SET(22,Y,4)
550 SET(39,Y,4)
560 NEXT Y
570 PRINT@340, CHR$(183);
580 FOR Y=16 TO 19
590 SET(41,Y,4)
600 NEXT Y
610 FOR X=14 TO 47
620 SET(X,25,2)
630 SET(X,28,2)
640 NEXT X
650 PRINT@423, CHR$(149);
660 PRINT@439, CHR$(154);
670 PRINT@424,"*SB-BAR BANDIT*";
680 PRINT@480,"PRESS SPACE BAR";
690 SB$= INKEY$
700 FOR GL=1 TO 8
710 PRINT@244, CHR$(149);
720 NEXT GL
730 FOR BL=1 TO 8
740 PRINT@496, CHR$( RND(10)+146
);
750 PRINT@244, CHR$(229);
760 NEXT BL
770 IF SB$="" THEN 690
780 IF SB$<>  CHR$(32) THEN 690
790 FOR WP=424 TO 438
800 PRINT@WP, CHR$(230);
810 NEXT WP
820 GOTO  150
830 D$="THREE OF A KIND PAYS $10
  "
840 W=10
850 GOTO  910
860 D$="..TWO OF A KIND PAYS $2
.."
870 W=2
880 GOTO  910
890 D$="*** JACKPOT *** PAYS $50
.."
900 W=50
910 FOR A=1 TO  LEN(D$)-10:E1=E1
+1:IF E1>4 THEN E1=1
920 PRINT@172, MID$(D$,A,10);
930 SOUND  RND(240),1
940 NEXT A
950 D$=""
960 P=W+LW:PRINT@109,P:LW=P
970 W=0
980 LW=LW
990 GOTO  690
```

Corrections

Frank Boyes has sent us this improvement to his program "DRAW" (July 1984 MiCo). Change line 480 to read:

```
480 IF A$="!" THEN 4000
```

This avoids the storage of GOSUB return addresses that will not be used, and avoids an out of memory report if you move several times from the re-draw routine back to the draw routine.

Also there is an undocumented feature within the blanking routine (using right slash '/').

If it is pressed several times without alternately using a direction key to move the point on the screen, it has the effect of removing the latest positions from memory. This can be seen and used by entering the re-draw routine (by using the minus sign "-").

BEAT THE HEAT

by Les Thurbon

In the interests of preserving the health of Gold Coast Rainbow editors and Queensland programers here are two utilities to enable you to determine how much they must suffer as they spend their idle days bending Bananas.

Listing 1, Temperature and Humidity Index is used to calculate the effective temperature on those hot humid days, enabling the correct adjustments to be made to the air conditioner. If you wish to care for your plants or age salami's in your kitchen then T.H.I. becomes invaluable.

Listing 2, Windchill has been designed to quickly determine the likelyhood of your suffering from exposure after standing for too long in the breeze from the airconditioner.

As can be appreciated, careful attention these two factors can make the environment we live in far more comfortable and in turn improve the way we work. In fact, anywhere there is a need for a controlled environment, why not simplify the task and let the computer do the mundane calculations.

Listing 1:

```
10 REM TEMPERATURE HUMIDITY
       INDEX.
20 REM BY LES. W. THURBON.
       COPYRIGHT CANBERRA 1984.
30 CLS0:PRINT@2,"AIR CONDITIONIN
G IS REQUIRED  IF THE T.I. EXCEE
DS 23. HALF THE POPULATION FEEL
DISTRESS WHEN  THE T.I. EXCEEDS
25. MISTAKES   AND ACCIDENTS IMC
REASE";
40 PRINT" AS THE   T.I. INCREASE
S TO 27. MANY      ANIMALS SUFFE
R AND ARE IN DANGER OF DYING WHE
N THE T.I. EXCEEDS  29. T.I. VAL
UES ABOVE 33 ARE    dangerous to
 most people";
50 FORA=1TO20000:NEXT
60 CLS0:SOUND200,10:PRINT@226,"T
EMPERATURE HUMIDITY INDEX!";:FOR
X=1TO3000:NEXT:CLS0
70 CLS0:PRINT@230,;:INPUT"HUMIDI
TY IN %";H:H=H/100:CLS0
80 PRINT@228,;:INPUT"TEMPERATURE
 IN (C)";T:T=(T*1.8)+32:T=T+.556
:CLS0
90 T=T-(.556-(.556*H)*(T-58.1))
100 T=INT(T)
110 IFT<60THEN190
120 IFT>150THEN180
130 IFT>90THEN240
140 IFT>84THEN230
150 IFT>79THEN220
160 IFT>75THEN210
170 IFT>60THEN200
180 CLS4:SOUND200,100:PRINT@235,
"*EVERYONE*";:PRINT@264,"*BUT YO
U IS DEAD*";:GOTO250
190 CLS0:PRINT@232,"***FEELS COL
D***";:GOTO250
200 CLS3:SOUND200,5:FORX=1TO5:PR
INT@231,"***COMFORT ZONE***";:FO
RY=1TO300:NEXTY:CLS3:FORS=1TO200
:NEXTS:NEXTX:GOTO250
210 CLS2:SOUND200,5:FORX=1TO5:PR
INT@230,"***SOME DISCOMFORT***";
:FORY=1TO300:NEXTY:CLS2:FORS=1TO
200:NEXTS:NEXTX:GOTO250
220 CLS8:FORX=1TO3:SOUND200,5:PR
INT@232,"***DISCOMFORT***";:FORY
=1TO300:NEXTY:CLS8:NEXTX:GOTO250
230 CLS7:FORX=1TO5:SOUND200,3:PR
INT@230,"heat stress warning";:F
ORY=1TO300:NEXTY:CLS7:NEXTX:PRIN
T@261,"***ACUTE DISCOMFORT***";:
GOTO250
240 CLS4:FORX=1TO10:SOUND200,3:P
RINT@228,"danger heatstroke aler
t";:FORY=1TO300:NEXTY:CLS4:NEXTX
:PRINT@261,"***ACUTE DISCOMFORT*
**";250 C=(T-32)*(5/9):C=INT(1*C
+.5)/1:PRINT@323,"TEMPERATURE HU
MIDITY INDEX";:PRINT@356,T"DEG.F
. OR ";C;"DEG.C.";:FORX=1TO10000
:NEXTX:GOTO70
```

Listing 2:

```
10 CLS0:SOUND200,10:PRINT@232,"W
INDCHILL FACTOR";:FORX=1TO3000:N
EXT:CLS0
20 CLS0:PRINT@230,;:INPUT"TEMPER
ATURE IN (C)";T:T=(T*1.8)+32:CLS
0
```

```
30 CLS0:PRINT@228,;:INPUT"WINDSP
EED IN (KMS/HR)";W:V=W/1.609344:
V=.44704*V
40 IFV<1.78816THENV=1.78816
50 A=10.45:B=10:C=-1
60 K=(A+(B*(SQR(V)))+(C*V))*(33-
T)
70 T=33-(K/22.034)
80 U=5/9*(T-32):U=INT(100*U+.5)/
100:T=INT(100*T+.5)/100
90 IFT<-19THEN150
100 IFT<-10THEN160
110 IFT<25THEN170
120 IFT<60THEN180
130 IFT<79THEN190
140 IFT>79THEN200
150 CLS4:FORX=1TO10:SOUND200,3:P
RINT@228,"danger frostbite warni
ng!";:FORY=1TO300:NEXTY:CLS4:NEX
TX:PRINT@261,"***ACUTE DISCOMFOR
T***";:GOTO200
160 CLS7:FORX=1TO5:SOUND200,3:PR
INT@231,"frostbite warning";:FOR
Y=1TO300:NEXTY:CLS7:NEXTX:PRINT@
261,"***ACUTE DISCOMFORT***";:GO
TO200
170 CLS8:SOUND200,5:FORX=1TO5:PR
INT@231,"WARNING VERY COLD";:FOR
Y=1TO300:NEXTY:CLS8:FORS=1TO200:
NEXTS:NEXTX:GOTO200
180 CLS0:PRINT@232,"***FEELS COL
D***";:GOTO200
190 CLS3:SOUND200,5:FORX=1TO5:PR
INT@231,"***COMFORT ZONE***";:FO
RY=1TO300:NEXTY:CLS3:FORS=1TO200
:NEXTS:NEXTX:GOTO200
200 PRINT@328,"WINDCHILL FACTOR"
;:PRINT@355,T" DEG.F. OR ";U;" D
EG.C.";:FORX=1TO10000:NEXTX:GOTO
20
```

HINT

Q. EXTENDED Basic EDIT commands
A. The following Edit commands are not shown on the Help card provided by Tandy...

EDIT10	allows editing of line 10
A	cancels all change, continues edit.
Q	cancels all change, ends edit
E	enters all changes
K	kills (deletes) rest of line
nK	kills up to the n th occurence of the specified character

Q. SLOW LIST
A. POKE 359,60 (ENTER).
Return...POKE(359,126)

MAKING YOUR OWN KIND OF

MUSIC

by Frank Boyes

COMPOSE allows you to compose a tune at your own pace, and at any time, replay the tune you have created.

The tune is composed by pressing the keys to play the notes. If required you can insert a silent space between the notes by pressing (Shift) A.

To enter the replay mode you press (Shift) 1.

Continue replaying the tune by pressing R or compose a new tune by pressing C.

The length of the notes is is controlled by line 1000 and line 3700 SOUND B,5 The 5 can be altered to suit.

Silent spaces are controlled by line 2000 FOR C=1to500 C is the control.

For audio visual effect see lines 3620 to 3680.

The Listing:

```
  1 REM *COMPOSE* BY FRANK BOYES
  50 CLEAR2000
  60 CLS
  100 GOSUB6000
  200 B$=""
  400 A$=INKEY$:IFA$=""THEN400
  440 IFA$="!"THEN3000
  500 B$=B$+A$
  600 B=ASC(A$)
  800 PRINTA$;
  900 IFA$=CHR$(137) THEN400
  1000 SOUNDB,5
  1200 GOTO400
  2000 FORC=1TO500:NEXT:NEXT
  2100 IFC$=CHR$(137) THEN4000
  3000 PRINT:FORMUS=1TOLEN(B$)
  3400 C$=MID$(B$,MUS,1)
  3500 PRINTC$;
  3550 IFC$=CHR$(137) THEN2000
  3570 IFC$=""THENPRINT"NO NOTES E
NTERED ** PRESS A KEY":PRINT:GOT
O200
  3600 B=ASC(C$)
  3620 REM************************
  3640 REM FOR AUDIO VISUAL EFFECT
  3650 REM CAN BE;-   COL=INT((B/3
2)+0.5)+1:CLSCOL
  3660 REM IF USING THIS, THEN
       DELETE LINE 3500
  3680 REM************************
  3700 SOUNDB,5
  3800 NEXTMUS
  4000 PRINT:PRINT"TO REPLAY THIS
TUNE, PRESS (R)"
  4040 PRINT"TO COMPOSE ANOTHER, P
RESS (C)"
  4500 D$=INKEY$:IFD$=""THEN4500
  4540 IFD$="R"THEN3000
  4560 IFD$="C"THENRUN
  4580 GOTO4500
  5999 REM START
  6000 CLS
  6020 FORA=100TO200STEP10:SOUNDA,
1:NEXT
  6050 PRINT"AS YOU PRESS A KEY, A
 NOTE WILL BE PLAYED,AND THE CHA
RACTER      DISPLAYED"
  6100 PRINT"EXCEPT THAT SHIFT 1 (
!) SENDS    THE PROGRAM TO A REPL
AY MODE"
  6140 PRINT"AND SHIFT A ("CHR$(13
7)") ALLOWS THE         INSERTION O
F A SILENT SPACE        BETWEEN NOT
ES"
  6180 PRINT,,"PRESS A KEY"
  6200 FORA=100TO200STEP10:SOUNDA,
1:NEXT
  6500 RETURN
  6900 REM SEE LINES 3620 TO 3680
```

FOR THE MATHEMETICIANS by Les Thurbon

Prolific Les Thorbon has submitted the four programs which follow.

The programs are designed to run on the MC-10 AND CoCo.

The machine used to write the programs is a 64K (A1) CoCo, the programs being saved to tape from the CoCo. This is done with a clever little program called 'CoCo-MiCo', an emulator program that fools CoCo into thinking that it is a MiCo.

See, many CoCo owners like the little machine so much that they have to have emulator programs just so they can experience the joys of working with MiCo!

AMORTIZATION SCHEDULE

The program is self-explanatory and runs correctly if typed in correctly but do take note of the period counter while the program is running for intermediate results.

To get the program to copy results to a printer, all PRINT @ statements in lines 290, 300, 310 AND 340 need to be changed to LPRINT (OR PRINT #-2 for CoCo). Some printers then require the additional line "LPRINT CHR$(13)" to produce a line feed.

The mathmatics is correct for equalised repayments but remember that many firms do not operate the equalized repayment agreement type of schedule. PERHAPS THEY TOO SHOULD PURCHASE AN MC-10 OR CoCo!

Listing 1:

```
10 POKE17026,6:CLS0:DATA37,69,10
1,133,165,166,167,168,169,171,45
,77,109,141,173,174,175,143,176,
177,145,113,81
20 DATA49,179,53,54,55,56,57,87,
119,151,183,185,-1
30 A$=CHR$(175):FORA=1TO35:READD
40 PRINT@D,A$;:FORT=1TO10:NEXT:N
EXT:PRINT@267," SOFTWARE ";:PRIN
T@456," CANBERRA  1984 ";
50 PRINT@326,"AMORTAZATION SCEDU
LE";:PRINT@395,"COPYRIGHT";
60 FORX=1TO5000:NEXT:CLS
70 X$=" ************************
******"
80 K=.5
90 INPUT" ENTER THE PRINCIPAL ";
D
100 INPUT" ENTER THE DEPOSIT ";Z
110 INPUT" ENTER THE ANNUAL INTE
REST RATE  IN PERSENT (%) ";R
120 IF R>1THEN R=R/100
130 PRINT:PRINT" ARE THE PAYMENT
S MONTHLY (M),   QUARTERLY (Q),
OR ANNUAL (A)"
140 INPUT" CALCULATING WILL BE F
OR THE     SAME PERIOD OF TIME "
;A$
150 IFA$="M" THEN C=12
160 IFA$="A" THEN C=1
170 IFA$="Q" THEN C=4
180 IFC=0THEN130
190 PRINT:INPUT" ENTER THE NUMBE
R OF PAYMENTS TO BE MADE";N
200 R1=R/C:CLS
210 PRINTX$:PRINTTAB(6)"AMORTAZA
TION SCEDULE":PRINTX$
220 D=D-Z
230 P=INT(D/((1-(1+R1)^-N)/R1)*1
00+K)/100
240 PRINT@102,"PRINCIPAL IS $";D
250 PRINT@136,"DEPOSIT IS $";Z
260 FORI=1 TO N
270 I1=INT((D*R1)*100+K)/100
280 PRINT@193,"PERIOD";I
290 PRINT@225,"PAYMENT $";P
300 PRINT@257,"INTEREST $";I1
310 PRINT@289,"TO PRINCIPAL $";I
NT((P-I1)*100+K)/100
320 D=INT((D-(P-I1))*100+K)/100
330 PRINT@321,"BALLANCE AFTER $"
;D
340 FORY=1TO5000:NEXTY
350 NEXTI:FORY=1TO5000:NEXTY
360 PRINT:PRINTX$:PRINT"      PR
OCESSING COMPLETE":PRINTX$
370 GOTO370:END
```

D.M.S. DEGREES. MINUTES. SECONDS OR (HOURS. MINUTES. SECONDS.)

Type in this short program and it wil convert decimal degrees or decimal time to D.M.S. or H.M.S.

Just enter your DECIMAL NUMBER. eg. 42.687 and your answer will be displayed with a decimal point separating each one.

TUTORIAL - Design a program using similar or the same mathmatics used in this program to convert D.M.S. or H.M.S. TO DECIMAL DEGREES OR DECIMAL TIME.

Listing 2:

```
10 CLS:CLEAR:PRINT:PRINT"  DEGRE
ES. MINUTES. SECONDS."
20 PRINT"   BY  L. W. THURBON. 1
984."
30 FORX=1TO5000:NEXTX
40 CLS:PRINT:INPUT"  ENTER DECIM
AL DEGREES";D
50 E=D-INT(D):D=INT(D)
60 E=E*60:F=E-INT(E)
70 E=INT(E)
80 F=(F+.01)*60:F=INT(F)
90 PRINT:PRINT"  D.M.S. =";D;"."
;E;".";F
100 FORX=1TO5000:NEXTX:GOTO40
110 END
```

CONVERGANCE TO PI. ( )

Type in this and run, then enter a number between .01 and 1000. The coputer will then converge to PI and will print "D ONE" when finished. This is only a small crude program for computing pi and so is only accurate to 7 DECIMAL PLACES IN THE MANTISSA due to truncation errors inherant to this computer.

Listing 3:

```
10 POKE49151,64:CLS0:DATA37,69,1
01,133,165,166,167,168,169,171,4
5,77,109,141,173,174,175,143,176
,177,145,113,81
20 DATA49,179,53,54,55,56,57,87,
119,151,183,185,-1
30 A$=CHR$(175):FORA=1TO35:READD
40 PRINT@D,A$;:FORT=1TO10:NEXT:N
EXT:PRINT@267," SOFTWARE ";:PRIN
T@456," CANBERRA  1984 ";
50 PRINT@328."CONVERGANCE TO PI"
;:PRINT@395,"COPYRIGHT";
60 FORX=1TO5000:NEXT
70 CLS:CLEAR:PRINT" ************
**********":PRINT@38,"CO
NVERGING TO PI."
80 PRINT"      TO 6 DECIMAL PLAC
ES.":PRINT" ********************
**********"
90 INPUT"ENTER A NUMBER <1000.";
A:IF A<.01 THEN A=.01
100 IF A>1000 THEN A=1000:B=90/A
:C=SIN(B/57.29577951)
110 D=C*2:E=D*A:A=A+E:E=INT(E*10
000000+.5)/10000000
120 PRINT@232,"PI =";E
130 IFE>3.1415926 THEN150
140 GOTO100
150 PRINT@301,"DONE";:GOTO150
160 END
```

Listing 4:

```
10 POKE49151,64:CLS0:DATA37,69,1
01,133,165,166,167,168,169,171,4
5,77,109,141,173,174,175,143,176
,177,145,113,81
20 DATA49,179,53,54,55,56,57,87,
119,151,183,185,-1
30 A$=CHR$(175):FORA=1TO35:READD
40 PRINT@D,A$;:FORT=1TO10:NEXT:N
EXT:PRINT@267," SOFTWARE ";:PRIN
T@456," CANBERRA  1984 ";
50 PRINT@326,"ARCTAN-ARCSIN-ARCC
OS";:PRINT@395,"COPYRIGHT";
60 FORX=1TO5000:NEXT:POKE49151,0
70 CLS:CLEAR:POKE17026,6:PRINT:P
RINT" enter 1 for arctan.":PRINT
" enter 2 for arcsin."
80 PRINT" enter 3 for arccos.":P
RINT" enter 4 for menu.":INPUT"
    ";Z:CLS:ON Z GOTO 220,90,150
,70
90 PRINT" press R to return to m
enu.":INPUT" enter   your number
 for arcsin      ";Y
100 X=Y/SQR(-Y*Y+1):GOTO230
110 CLS:PRINT@226,"arcsin of";Y;
"= ";C;"deg.":PRINT:PRINT"  arcs
in of";Y;"= ";A;"rads."
120 A$=INKEY$:IFA$<>"R" THEN120
130 IFA$="R" THEN70
140 END
150 PRINT" press R to return to
menu.":INPUT" enter   your numbe
r for arccos.     ";Y
160 X=Y/SQR(-Y*Y+1):GOTO230
170 A=1.570796327-A:C=90-C:CLS:P
RINT@226,"arccos of";Y;"= ";C;"d
eg."
180 A=INT(A*10000+.5)/10000:PRIN
```

```
T:PRINT"  arccos of";Y;"= ";A;"r
ads."
190 A$=INKEY$:IFA$<>"R" THEN190
200 IFA$="R" THEN70
210 END
220 PRINT" enter R to return to
menu.":INPUT" enter   number for
 arctan.          ";X
230 IFX<0THEN T=-1
240 IFX>0THEN T=1
250 X=ABS(X):C=0
260 IFX>1THEN280
270 GOTO290
280 C=1:X=1/X
290 A=X*X
300 B=((2.86623E-3*A-1.616553E-2
)*A+4.29096E-2)*A
310 B=((((B-7.5289E-2)*A+.106563
007)*A-.142089)*A+.199936189)*A
320 A=((B-.333333333)*A+1)*X
330 IFC=1THEN350
340 GOTO360
350 A=1.570796327-A
360 A=T*A
370 C=A*57.29577951:C=INT(C*1000
0+.5)/10000:A=INT(A*10000+.5)/10
000
380 IFZ=2THEN110
390 IFZ=3THEN170
400 CLS:PRINT@226,"arctan of";X;
"= ";C;"deg.":PRINT:PRINT"  arct
an of";X;"= ";A;"rads."
410 A$=INKEY$:IFA$<>"R" THEN410
420 IFA$="R" THEN70
430 END
```

As I said (in June's issue of MiCo) you can use RaNDom provided you play by the rules. In this example RaNDom is inside CHR$() statement parenthesis. Enter this example:

```
10 CLS
20 PRINT@200,CHR$(RND(225))
30 SOUND 100,3
40 GOTO 20
```

Here line 20 prints a character but a RaNDom CHR$() from 225. Line 30 is used to hold the character on the screen (the computer sounds 100,3 before going to 20) then prints another RaNDom CHR$(). Suppose I don't want numbers and letters, just graphic characters. Looking up the manual for the MC-10 (pg 118 -:for CoCo pg. 276 of Getting Started) we find that graphics characters range from 128 to 143 and to create a character in colour you add the appropriate number to the code. For example PRINT CHR$(129+16) produces a yellow character 129. This adding the 16 to produce a yellow charater only uses extra memory. Do the adding for MiCo and save memory. For example 129+16=145 now type PRINT CHR$(145). The answer is the same. Take character No. 143 (the last character on page 118) then add 112 to get the last colour orange and the answer is 255. Now we know the graphic CHR$() range from 128 to 255 but if I ask the computer to PRINT RND(255) it will pick a number from 1 to 255 and I'm back to numbers, letters and characters. Here is a way to keep the CHR$(RND()) to a given range. The graphic CHR$() range from 128 to 255. The characters we don't want are those from0 to 127. Now type the following:

```
10 CLS
20 PRINT@200,CHR$(RND(128)+127)
30 SOUND 100,3
40 GOTO 20
```

Now RANDOM stays in the range of graphic CHR$(). Type this small program:

```
5  CLS
10 FOR X=1 TO 20
20 PRINT@200,CHR$(RND(128)+127)+
CHR$(RND(128)+127)
30 PRINT@232,CHR$(RND(128)+127);
CHR$(RND(128)+127)
40 NEXT
```

Notice how I kept adding CHR$(RND(128)+127). You could make a string of these for some special effect or just flicking lights.

...until next month keep RANDOM rolling.

Continued from P 29.

Harold ratchetted the handbrake to hold the car, pushed sideways to open the vehicle's door, and, creaking his neck, he peered upwards through straining eyes tired from the long hours of concentration at the computer keyboard. ... There was a face up there, all right ... now pressed flatly against the glass. Harold raised his arm, seemingly not knowing whether to menace the owner of the face as a malefactor or to grudgingly greet the unknown with weariness, witless welcome, empty and unmeant. Halfway up, his hands flapping half-heartedly, still patently uncertain ... with a flash, Harold remembered ... somthing Jane had said ... and that he'd only half heard.

The eyes above the high cheek bones, wide open like a child's in wonderment, or ... was it pleading?

Then, without warning, the face slid down the window and as it did, a hand, long fingered and thin, clasped at the inner window ledge, and, as if its owner had lost strength suddenly, the fingers opened - and with a gentle thud, Harold heard the body upstairs flop to the floor.

Harold remembered ...
He ran rapidly to the front door of the house and tested it. Locked, of course. He'd seen the carriers using the owner's deadlock keys just before they'd left with the giant pantechnicon loaded on their way to their furniture repository. ... But the Newlings had sold the place fully furnished. Harold could'nt stop his brain programming that ... and other ... memories. ... The removalist's van ... The furniture ... And what Jane had said ... he'd remember that too, later.

No way of getting into the Newling's house. Only thing to do now was to call the police. ... Harold's mind kept going back to the fact that the place was to have been sold fully furnished. Jane had discussed this with Harold at the time - well, half discussed it - when the van had come to take all the contents away. But she had shrugged it off. 'They must have changed their minds' Jane had affirmed. And Harold had accepted her verdict as just sheer feminine logic, which of course he must admit would have been assisted by his clearly indicated offhandedness about the whole affair.

But right now, Harold kept to his decision. He dialled the local police station.

"Sergeant-in-charge?" he asked.
He must have got an affirmative answer, for he went on quickly.

"I...want to report...a theft..."
Harold couldn't help hesitating before he added:
"...A houseful of furniture, Sergeant Jackson."

Harold suddenly recalled the name, which he should well have known, as they'd met on several occasions at a local club.

Harold knew there'd be an explosion at the other end when he answered the Sergeant's obvious question.
"...When did this happen, sir?"
with
"Ten days ago."

The police sergeant cursed crisply for a full minute without repeating an expletive. Then he recovered rapidly.
"... And you left it until now to report?"

Harold could feel the strained, stiffled patience, in the sergeant's voice.

Then when Harold explained about the face, or faces, at the window, there was a louder uproar from Sergeant Jackson. Harold held the telephone away from his ear until it subsided.
Then he heard:
"... I'll be over right away, with the paddy-wagon and a couple of my men."

Harold caught the click of the receiver at the other end as it was placed in its cradle.

He heard steps in the room and looked around.

... Jane had come in when she'd heard

Harold speaking on the phone after rattling through her kitchen without his usual greeting.

"What's so urgent?" she asked, with a brittle, almost sarcastic, tone. All she said, when Harold explained, was:

"Oh."

... Then she went back into her kitchen. It was as if what he had said was about a casual, everyday occurrence ... or was it an acceptance of something that was already known to her, but not to him?

Harold did not notice his wife's offhand attitude. Preoccupied with the current crisis in which he was enmeshed, Harold hurried out to take one more look up at that window, where the faces had appeared, before the police arrived. He was sure now that he had seen more than one face.

... Harold looked up at the window. The two faces were there again, both mutely pressed against the glass. It appeared that they were kneeling, Harold at last realised. He hadn't thought of it before. Why would they be kneeling? The two pairs of upstretched hands waved, beckoned to Harold. Why? Who were they? ... He watched as the hands wavered, and as the faces stiffened, both slid out of sight.

... The big Ford paddy-wagon pulled up across the driveway of the vacant house, and Sergeant Jackson and his two constables got out.

One of the men placed himself at the rear of the vehicle, unlocked the doors, propped them open, and with folded arms he stood ... ready to receive. The other ran around to the back of the house. Sergeant Jackson went to the front door.

It was then that Harold saw the keys dangling from a thin chain held in the sergeant's pudgy left hand. Sergeant Jackson hadn't told Harold that the Newlings had left a set of house keys both with him and with the estate agent before they had gone overseas. And that the sergeant remembered, but Harold didn't, that Harold had promised Sergeant Jackson and the Newlings that he'd keep an eye on the place, and report anything untoward. ... Untoward? ... A missing houseful of furniture. ... But that wasn't all.

... Sergeant Jackson and his senior constable brought out the two emaciated, knee-bound figures, Harold's mind became freed from the self delusion of the past ten days. Remembering what Jane had done, what he had heard and seen, Harold asked Jackson to wait.

"Wait?" questioned the police officer querulously. "Wait for what? These people can't wait..." He pointed angrily at the two mute, senseless people who had been taken from the empty house. It was evident that apart from the restriction of their bonds, they were also affected by drugs.

"A quick word with my wife---" requested Harold anxiously.

"Hurry, then."

Harold went looking for Jane. As he walked along to the side entrance of his home his eyes were compulsively drawn to the window of the house next door... The faces were there again, gesticulating, pointing downwards, jabbering and giggling wordlessly. ... But they could'nt be there - the two people were with the police, at the paddy wagon, too stoned to talk!

Harold found Jane in the living room. She turned guiltily as he entered. He saw her jab at the keys and jolt down the switches of their CoCo and video player. ... He had a flash of intuition. Turning quickly about, Harold dashed back to the door and looked up again. The faces ... were gone from the window! ... Peering intensely, Harold caught the disappearing pin point on the screen where next-door's window glass had been blanked out. ... A grotesque computer game?

Jane explained. She had tried before, but Harold hadn't listened. Perhaps that's why all this happened!

"When the Newlings sold the house, the settlement cheque bounced and they couldn't go overseas. They asked me to sell the furniture for them so that they could go anyway. ... And we rigged up a screen so that you ... everybody ... would think that someone was still there."

Harold interrupted; "But who are the people the police found in there? ... And how did they get gagged and bound and drugged?"

"... It's a long story ..." Jane said lamely. "... It's all on the video."

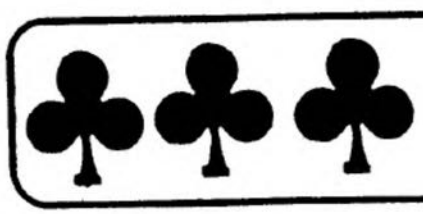
She went back to the machine, tapped at the keys, flipped the switches, and with a whirr the picture came into focus. Words and pictures ... Harold watched the story unfold. ... The Newlings waving a worthless cheque, gesticulating at the estate agent.

... The removalist's van taking out all their belongings at midnight.

... The two mysterious figures ushered into the house, gagged, bound and drugged. ... Then, himself ... watching the window ... calling the police. The police! Harold rushed outside ... No police. ... No van ... No people. He looked up again at the blank window as he went back to rejoin Jane.

"Play it again, Jane," he chuckled. "... Before I go back to work on a real computer!"

HARRY WALLIS



CLUB NEWS

MEETS.

Many of our meet groups are very successful in getting people along and keeping them interested. Others are not so successful, and still others worry about whether they have enough numbers, etc.

When you decide to hold a meet you are dealing with Joe Public.

Let me tell you about Joe Public.

Negatives:

1. He is self centred.
2. He is self centred.

Positives:

1. If you show you care, you've got an 80% chance of winning him if you don't pressure.

I've always had the philosphy that if we hold a meet, and two turn up and enjoy the meet, then that was a good meet. I don't believe in going for numbers. I'm lazy and lots of people means lots of work for someone - usually me. I wouldn't discourage large numbers, but when a meet gets to about 15 people, its growth rate starts to drop off and to get regular attendances of about 30 is exceptional. (And yes, I know, there ARE several exceptional meets around). In fact, there are good psychological reasons why you get number drop off at about 15 people. The incidence of this type of occurrence is quite well documented.

What happens is that in small meets, whether you want to or not, you've almost GOT to talk to everyone there. In groups of 15 onwards, the positive is that if there's a new person there, or someone you don't particularly like is there, then you can get away without talking to them. I said the positive - that's also the negative, because that guy is unlikely to come back.

So here's a couple of rules to observe:

1. When you get to 15 members, seriously consider splitting the group.

2. Think about the people who are at the meeting. Ask yourself, "Why are they here?" Then ask them the same question. Then do something about it!

3. There's no substitute for hands on experience, where a computer is concerned, so have as many computers at the meeting as possible.

4. Remember how you felt the first time you turned up at a meet. Some people could be even shier than you!

5. Don't worry about numbers - have fun with whoever turns up!

I don't believe it!

I complained last month that the Latrobe Valley User Group's newsletter kept disappearing from the office, it turned up, (as I thought, Kevin had it!), I hid it so that I could refer to it now, and would you believe that it has gone again!

Great little newsletter that - if you ever get to read it.

CoCoPug, the newsletter of the Perth users group, has a number of nice little programs in it. I particularly like the look of "Submarine Destroyer", a program modified from one originally written for the Model 1.

Perth is just about to get their Bulletin Board off the ground. Like us, the phone number hasn't been allocated, and they will probably have quite a few teething problems, so they are wisely limiting calls at this stage.

CoCoBug from the Brisbane group came again this month. It has a program that gives you 10 colors in PMODE 3, a review of the AMUST printer, Splat, and

something really interesting, "Star Trek - The Adventure". There is also a graphics screen manipulator, and several other good quality programs. Not a bad start at all for new Editor, Barry Cawley.

Lindsay Wrigley has become a contact for New Zealand. He lives in Lower Hutt, and will also act as agent for your software needs.

Ivor Davies, (Melbourne) is one of a number of people, particularly in Telecom it seems, who are involved in Users Groups at work. His group has just started, and concentrates logically enough, on hardware.

Peter Miller, the nice guy of Maryborough, has had to return to Melbourne, due to family health problems. In his place stands Norm Wynn (071 68 1628), who is looking forward to starting a meet in the new year.

Jeff Larsen phoned from Cairns, he and the local Tandy agent, Tom Booy are planning the first meet of the Cairns User Group on 16th December, at 1PM. Contact is Tom on 070 51 1795.

That's it for this month. Keep that information flowing in - we want to know what you're doing.

---

uuuuser groupppppp
ddCONTACTSSSS

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

ADELAIDE	JOHN HAINES 08 278 3560
ADELAIDE NTH	STVN EISENBERG 08 258 6214
ALBURY	RON DUNCAN 060 43 1031
BAIRNSDALE	COLIN LEHMANN 051 57 1545
BALLARAT	MARK BEVELANDER 053 32 6733
BANKSTOWN	KEN HAYWARD 02 759 2227
BLACKTOWN	KEITH GALLAGHER 02-627-4627
BLAXLAND	BRUCE SULLIVAN 047 39 3903
BOWEN	TONY EVANS 077 86 2220
BRASSALL	BOB UNSWORTH 07 201 8659
BRIGHTON	GLENN DAVIES 08 296 7477
BRISBANE EAST	ROB THOMPSON 07 848 5512
BRISBANE NTH	JACK FRICKER 07 262 8869
BRISBANE STH	PATRIC SIMONIS 07 209 3177
BRISBANE SW	GRAHAM BUTCHER 07 376 3400
BRISBANE WEST	BRIAN DOUGAN 07 30 2072
BUNDABERG	JIM McPHERSON 071 72 8329
CAMBERWELL	TONY BALDWIN 03 728 3676
CAMPBELLTOWN	LEO GINLEY 02 605 4572
CANBERRA	SHAUN WILSON 062 51 2339
CARLISLE MiCo	STUART HALL 361 1922
CAULFIELD"?526..	"JEFF SHEEN 03 528 3724
CHATSWOOD	BILL O'DONNELL 02 411 3336
CHURCHILL	GEOFF SPOWART 051 22 1389
COLYTON TEENS	DWAYNE MANSON 02 623 5005
COOMA	SHEILA HAMMILL 064.82.3905
DANDENONG	BRETT CRUICKSHANK 03 547 6604
DARWIN	BRENTON PRIOR 089.81.7766
DENILIQUIN	WAYNE PATTERSON 058 81 3014
DUBBO	GRAEME CLARKE 068 89 2895
EMERALD	CAROL CATHCART 059 68 3026
FORSTER	GARY BAILEY 065 54 5029
FRANKSTON	BOB HAYTER 03.783.9748
GIPPSLAND STH	PAT KERMODE 056 74 4583
GOLD COAST	SHERYL BENTICK 075-39-2003
GOSFORD	PETER SEIFERT 043 32 7874
GRAFTON	DAVID HULME 066 42.0627
GREENACRES	BETTY LITTLE 08 261 4083
HOBART	BOB DELBOURGO 002 25 3896
IPSWICH	MILTON ROWE 07 281 4859
JUNEE	PAUL MALONEY 069 24 1868
KENMORE	GRAHAM BUTCHER 07 376 3400
LEETON	CHRIS NAGLE FARM 769, MUSCAT ST AMESBURY, LEETON 2705
LITHGOW	STUART RAYNER 063 51 4214
LIVERPOOL	LEONIE DUGGAN 02-607-3791
MACKAY	LEN MALONEY 079511333x782
MACLEOD	ROBIN ZIUKELIS 03 450211x465
MacQUARIEFIELDS	KIETH ROACH 02 618 2858
MAFFRA	MAX HUCKERBY 051 45 4315
MAITLAND	LYN DAWSON 049 49 8144
MARYBOROUGH	PETER MILLER 071 22 1899
MELBOURNE	JEFF SHEEN 03 528 3724
MELTON	MARIO GERADA 03 743 1323
MILDURA	DOUG MATTHEWS 050 23 5701
MOE	STEPHEN SEMPLE 051 27 6841
MORWELL	GEORGE FRANCIS 051 34 5175
MT ISA PAUL	BOUCKLEY-SIMONS 077 43 6280
MUDGEE	BRIAN STONE 063-72-1958
NAMBUCCA HDS	WENDY PETERSON 065 68 6723
NEWCASTLE	LYN DAWSON 049 49 8144
NOWRA	ROY LOPEZ 044 48 7031
PARKES	DAVID SMALL 068 62 2682
PENRITH	TON LEHANE 047-31-5303
PERTH	JOHN CHRISTOU 09 344 6745
PORT MacQUARIE	RON LALOR 065 83 8223
PORT NOARLUNGA	ROB DALZELL 08 386.1647
PORT PIRIE	KEVIN GOMAN 086 32 1368
RINGWOOD	ANDREW RAWLINGS 03 726 6521
ROCKHAMPTON	KEIRAN SIMPSON 079 28 6162
ROCKHAMPTON MiCo	TIM SHANK 079 28 1646
ROSEVILLE	KEN UZZELL 02 467 1619
SALE	BRYAN McHUGH 051 44 4792
SINGLETON	DAVID NICHOLS 065-73-1222
SPRINGWOOD	DAVID SEAMONS 047 51 2107
STURT	MARY DAVIS 08 296 7477
SUNBURY	JACK SMIT 03.744.1355
SUTHERLAND	IAN ANNABEL 02 528 3391
SWAN HILL	BARRIE GERRAND 050.32.3838
SYDNEY TEENS	ROD HOSKINSON 02 48 5948
TAMWORTH	ROBERT WEBB 067 65 7256
TONGALLA	TONY HILLIS 058 59 2251
TOOWOOMBA	MIKE CARTER 076 35 6911
' BEGIN NTH	DAVID PROUT 076.32.7533
' BEGIN STH	LEW GERSEKOWSKI076 35 8264
' ADVANCED	GRAHAM BURGESS 076 38 4259
TOWNSVILLE	JOHN O'CALLAGHAN 077 73 2064
TRARALGON	MORRIS GRADY 051 66 1331
UPPER HUNTER	TERRY GRAVOLIN 065 45 1698
WAGGA WAGGA	BRUCE KING 069 25 3891
WESTLEIGH	ATHALIE SMART 02 848 0938
WHYALLA NORRIE	CHRIS HUNTER 086 45 3395
WOLLONGONG	BRIAN McCAULEY 042 71 4265
WONTHAGGI	PAT KERMODE 056 74 4583

059 Bulletin Board System:
' 9pm to 9am MODEM, 9am to 4pm VOICE
' SYSOP BOB THOMPSON 047 38 2468

CLASSIFIEDS

FOR SALE.

DMP 100 + New Ribbon. Good Condition.
$159.00 Phone 062 51 4074.
M. TURK. 3 Lochbuy St., MacQuarrie.
ACT. 2614.

OS-9 Recently purchased and little used.
Complete with Manuals and Packaging. $80.00.

Phone 03 751 1425
Ron Hutchins
Lot 4 Range Rd
OLINDA. VIC.

SOFTWARE SPECTRUM

correspondence: **P.O. Box 2101, Adelaide, South Australia, 5001.**

deliveries: **25 Selby Street, Adelaide**
Phone: (08) 211 8763 or 51 4868

TRS 80® COLOR COMPUTER SOFTWARE SPECIALISTS

Serial to Parallel Printer Interface Switchable from 600-9600 BAUD $99.95 Non-switchable 4800 BAUD $69.95
HJL-57 Professional Keyboard $139.95 5¼" Disk Drives 40-Track with Controller $599.95
3" Disk Drives 40 Track with Controller $529.95
BMC-FX Printer with Interface and Word Processor $469.95

SOFTWARE

Learn to Read Music	$47.95	Musica	$47.95
Graphicom	$40.95	Co Co Dump	$37.95
Co Co Checker	$37.95	Super Forth	$63.95
Tut's Tomb	$35.95	Black Sanctum (Graphics)	$35.95
Candy Co	$41.95	Erland	$35.95
Pengon	$35.95	Lunar Rover Patrol	$35.95

All products come with an unconditional Guarantee. If defective return within 14 days for replacement. **Dealer inquiries invited**

SEND $1.00 POSTAGE FOR CATALOGUE

To: SOFTWARE SPECTRUM, Box 2101 GPO, Adelaide 5000 **New Zealand Bankcard welcome**

Mr/Mrs/Miss/Ms ..... Address ..... Postcode .....

Phone ..... Please charge Bankcard No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Exp. Date ..... Sig. .....

NORTH QLD. COLOUR SOFTWARE
9 DURHAM COURT, KIRWAN, TOWNSVILLE 4814.
WE'RE USER FRIENDLY (077)732064

MEMORY 64K UPGRADES $89 ONLY

****
WRITE OR PHONE FOR MORE INFORMATION
****

(32K) EIGHT BALL EIGHTBALL EIGHT BALL $28-95
*TOTALLY REALISTIC*CONTROL THE CUE*
*HIT HARD,SOFT,MEDIUM*SCREW BACK,RUN ON*
SPIN & SWERVE /click/CLICK/go the balls!!!

so realistic you'll be looking for the chalk! SORRY!! NO SMOKING OVER THE TABLE!

"Your turn EDDIE!"

if not already on 64K, take a 5% discount on upgrade/eightball package.

32K*32K*32K*
VIP WRITER $6295/VIP DATABASE
DONKEY-what a classic! $2995
CASHMAN-40-yes, 40screens. Two players on screen at same time. Beat your opponent to the MONEY.
SPACE SHUTTLE-great! Realistic simulation. retrieve satellite, land on 3D runway,

64K UPGRADES ONLY $89 /KIT

16K

whirlybird-5 screens-guide your chopper on a mission- if you have the nerve! $2895
astroblast-defend your home base-try to refuel $2895
ghost gobbler- how's your blood pressure?-colourful PACMAN type. $2895

FILEMASTER-powerful database-speedy filing and sorting/run to prntr. $3420

AUSTRALIAN RAINBOW

The only other magazine you need for your TRS 80 Color Computer

AUSTRALIAN "CoCo
Registered by Australia Post-
Registration No. QBG 4007
Graham Morphett, P.O. Box 1742, Southport. QLD. 4215.

POSTAGE PAID AUSTRALIA