



SYDNEY TRS-80 USERS GROUP NEWSLETTER

P.O. BOX 297, PADSTOW 2211.

Volume. 6 Issue. 3 November 1985

| <u>Title</u> | <u>Author</u> | <u>Page</u> |
|---|------------------|-------------|
| Editor's Soapbox | Gary Bryce | 2 |
| Secretary's Sayings | Darrell Hegarty | 2 |
| Arranger on the Model 4P | Gary Bryce | 2 |
| Variable Storage | L.J. Lawes | 3 |
| Alan's Sector | Alan Morrison | 5 |
| Uploading from MICRODOS | Brendan Thompson | 5 |
| The Comm Line | Michael Cooper | 6 |
| SuperSCRIPSIT Patches | P. Kaggs | 8 |
| Repair SuperSCRIPSIT Files | Bits & Bytes | 9 |
| Visicalc Business Forecasting Model | Phil Walsh | 9 |
| LDOS / TRSDOS Corner | Gary Bryce | 11 |
| Using Unloadable Disks on TANDON Drives | Robert Jarrett | 12 |

Meeting News

As always the first and second monthly meetings will be held at the rear of Pattersons Florists, Botany Rd, BOTANY (entrance from Chegwyn St) while the third meeting (on the fourth Saturday of the month), will be held at the 1st Sefton Scout Hall, 2 Waldron Rd, SEFTON, all meetings commence at 1 PM on the following dates :-

| | | | |
|---------------|--------|-------------------|--------------------------|
| November 9th | Botany | December 14th | Monthly General Meeting |
| November 16th | Botany | December 21st | Special Interest Meeting |
| November 23rd | Sefton | December 28th ??? | South West Meeting |

Who's Who

| | | |
|-----------------------|-----------------|----------|
| President | Ted Romer | 498-2399 |
| Vice President | Jim Whittaker | 772-2009 |
| Treasurer | Greg North | 635-1438 |
| Secretary | Darrell Hegarty | 560-9681 |
| Newsletter Editor | Gary Bryce | 628-5058 |
| Club-80 Sysop | Michael Cooper | 331-7136 |
| CBBS Secretary | Peter Wignell | 759-8024 |
| Hardware Co-ordinator | Errol Rosser | 709-7646 |

SYDTRUG Bulletin Board

CLUB-80 Bulletin Board operates for members, seven days a week twenty four (24) hours a day on (02) 332-2494. The data format used is as follows :- 8 data bits, 1 stop bit, No parity, Full duplex, CCITT V21 modem standard 300 bps (set your modem to ORIGINATE mode). Limited access is granted for visitors.

The contents of this publication are (c) 1985 by SYDNEY TRS-80 USERS GROUP. All rights reserved. Enquiries should be directed to the "Secretary, Sydney TRS-80 Users Group, P.O. Box 297, PADSTOW N.S.W., AUSTRALIA 2211. Material appearing in this publication may be reprinted in similar computer club newsletters and nonprofit publications if accompanied by the following notice:

Reprinted from SYDTRUG NEWS, P.O. Box 297, PADSTOW 2211.

Editor's Soapbox

by Gary Bryce (02) 628-5058

Now that you have got past the front cover you should noticed the different format of the newsletter. Much as it saddens me, I have had to give up using "DOTWRITER" to format the complete newsletter. The time involved in formatting it each month was getting ridiculous (50 to 80 hours per issue), particularly when any articles contained "Tabular Information". This issue has been formatted and printed using "ALLWRITE", which though the formatting commands are very similar, does have some considerable advantages (i.e. printing both columns simultaneously, Tabs, print preveiwing, and a considerably faster throughput, to name just a few!!).

As you thumb through, you will also note that this issue is a bit shorter than usual. There is one good reason for this!! Yes, I'm sorry to say, the "Hobby Horse" is out again, the newsletter needs YOUR contributions!!!!

As our printer is going on holidays, this issue is being prepared almost two weeks early (we may do an early posting as well, particularly the way the postal situation is here in SYDNEY at the moment!), so this may explain the lack of contributions this month.

As I write this there are only nine weeks to Christmas (Jack Decker and our other Northern Hemisphere readers must envy the Sun, Surf and Sand), so you had all start being good boys and girls if you want "SANTA" to include any new "Gear" in your Christmas stocking.

So goodbye to all for this month, and "Keep on Computing".

Secretary's Sayings

by Darrell Hegarty (02) 560 9681

Not really very much to report to you all this month, so I'll be realistic, and only write what I have to report. Also, Gary has just told me that our printer is going on holidays, and the deadline has been bought forward by one week - this is Friday night, and I have to have this article ready by to-morrow's meeting at Botany. (That managed to fill up some lines).

The financial report this month will be of only the barest details, as the new treasurer had some trouble with the Visicalc disks giving parity errors.

The bank balance stands at \$7746.23 plus \$20.52 in an ANZ account which has been unused for some time. The income for the month of September was \$646.00, and the expenditure was \$1130.00. This seemingly large unbalance was mainly due to \$780.00 being spent to purchase Flippy disks for the distribution of the public domain software. The ANZ bank account mentioned is to be closed, and \$5000.00 of the funds are to be invested in some form of

interest-bearing deposits with a Building Society in order to earn some interest for the club.

A new printer has been purchased for the Bulletin Board, as the original printer (a TANDY LP VII) finally threw in the claw, and apparently cannot be repaired without sending it to TANDY (with an uneconomic repair cost).

The Public Domain software will be distributed on FLIPPY disks, each side of 35/40 tracks, SINGLE density and the catalogue disk is available at Botany for anyone to copy. There is a program on the catalogue disk to print an order form for mail orders, and the catalogue will be printed in the newsletter as each release of software is made. It is intended to release as limited number of disks each month, so that the whole lot will be released over about 6 months (to give the members who are handling the duplication and distribution a chance at handling the job).

The price for the disks will be \$10.00 each, with a 50% DISCOUNT to club members, plus \$1.00 postage per order. Please note that we are NOT selling the software, but the disks, plus a small amount for copying and handling. That makes it \$5.00 per disk to members, or \$10.00 per disk for non-members (plus postage of course).

We have a double-density board, suitable for the SYSTEM/80, and Model 1's (with the un-buffered interface cable), to be a prize in a "guessing competition" to be run via the newsletter, probably commencing next month, and going for one or two months. The value is about \$150.00, so it is a worthwhile prize for those of you who don't already have DD going.

Well - I think I found quite a bit more to say than I first anticipated, so that's all for now -----
Regards to all - Darrell H.

Arranger 2.1 on the Model 4P

by Gary Bryce (02) 628-5058

The "ARRANGER" by Dan Foy is in my opinion one of the best disk indexing programs available for the TRS-80. It will "BOOT" and operate in unmodified form on the Model 1, III and 4, but it will not "BOOT" on the 4P.

Why? The MODEL III ROM must be available prior to booting, as the ARRANGER operates in Model III mode on a Model 4 or 4P, and this means that the MODEL4/III file must be operational on a 4P before the ARRANGER can be loaded. You may, if it doesn't eventually give you a pain in the neck, use the Model4/III file disk followed by the ARRANGER, or execute the procedure detailed below to copy the ROM image to a backup of your ARRANGER master disk. The resulting disk can still be used on a Model 1, III or 4 without any problems.

1. Build the following patch file on drive 0.

.ADIR/FIX

.Patch to "LOCK OUT" all tracks below track 35

```
.
D00,00=FF FF FF FF FF FF FF FF
D00,08=FF FF FF FF FF FF FF FF
D00,10=FF FF FF FF FF FF FF FF
D00,18=FF FF FF FF FF FF FF FF
D00,20=FF FF FF
D00,60=FF FF FF FF FF FF FF FF
D00,68=FF FF FF FF FF FF FF FF
D00,70=FF FF FF FF FF FF FF FF
D00,78=FF FF FF FF FF FF FF FF
D00,80=FF FF FF
.End of Patch
```

2. Mount an unformatted disk on drive 1 then create and execute the following JCL.

```
.ARRANG4P/JCL
.
.Format a disk in drive 1 with the Directory on Trk 35
FORMAT :1 (NAME="ARRANGER",DIR=35,Q=N,ABS)
.
.Lockout Tracks below Track 35
PATCH DIR/SYS.LSIDDS:1 ADIR/FIX (Q=N)
.
.Correct the Directory Data Address Mark
REPAIR :1
.
.Copy the MODEL4/III file from drive 0 to drive 1
COPY MODEL4/III:0 :1
//EXIT
```

3. Mount the MODEL4/III File disk in drive 0 and press RESET
4. After the Model III ROM image has loaded, replace the MODEL4/III with your Master ARRANGER disk and again press RESET.
5. Follow the normal initialization procedure and select the BACKUP ARRANGER option from the menu to create a copy of the ARRANGER on drive 1.
6. Replace the ARRANGER disk on drive 0 with the TRSDOS 6.x disk and press RESET.
7. Complete the process by applying the following direct patch from TRSDOS READY :-

```
PATCH BOOT/SYS.LSIDDS:1(D01,00=00 31 23:F01,00=F3 31 00)
```

You may now move the diskette from drive 0 to drive 1 and press RESET. The Model III ROM image will load followed by the ARRANGER.

Please note any backups of the newly created disk using the Backup option of the ARRANGER will not include all the additions required (ie: Directory and MODEL4/III file) and therefore the above procedure must be repeated for all subsequent copies which are to be used on the 4P.

This procedure should work for many "Self Booting" Model III programs that you want to operate on the 4P. Some possible conflicts could arise with programs that use all forty tracks, but these should be rare (and if you are able to use an eighty track drive as drive 0, the directory could be placed above track forty). As these "Self Booting" programs must have a Boot Sector readable by the normal ROM routines, accessing sector 1 for modification should not be a problem.

Variable Storage

by L. J. Lawes

from "TRS-80 SYSTEM-80 Computer Group Newsletter"

One of the areas of computer study in which I have become interested is how the computer stores variables internally in memory. In researching the subject I went through the notes presented by Fred Malone when he spoke on this subject at a club meeting about three years ago. I also acknowledge as part of my research:-

TRS-80 HACKERS HANDBOOK for NEWDOS/80

by Kevin O'Hare

TRS-80/Z80 ASSEMBLY LANGUAGE LIBRARY

by Craig Lindley

The objective of my study was to see how TRS-80 stores its variables internally. The procedure I adopted was to write a short programme in BASIC using the different types of variables. Then I printed a HEX dump of the memory affected by the RUNNING of the programme, for closer examination. In this article, I will only be looking at simple Integer and Single Precision variables. Let us now go step by step through the process.

The Appendix is the LLISTING of the target programme, and the results of the variables where Random numbers were used.

The first step was to RUN the BASIC programme. Next, I needed to know where in memory to look, so I referred to the HACKERS HANDBOOK at pages 18-19 for the following information.

| ADDRESS | POINTER TO:- |
|-------------------------|---------------------------|
| 40A4-40A5 (16548-16549) | Start of BASIC |
| 40F9-40FA (16633-16634) | Start of Simple Variables |
| 40FB-40FC (16635-16636) | Start of Array Variables |
| 40FD-40FE (16637-16638) | Start of Free Memory |

From the Command Mode, I executed the Commands:-

```
?PEEK(16548)+PEEK(16549)*256 = 27206 = 6A46H
?PEEK(16633)+PEEK(16634)*256 = 28049 = 6D91H
?PEEK(16635)+PEEK(16636)*256 = 28103 = 6DC7H
?PEEK(16637)+PEEK(16638)*256 = 28418 = 6F02H
```

Using a monitor (in this case I used RSM) I obtained a HEX dump of the memory area 6A46-6F02. The programme itself resides in the area 6A46-6D90. BASIC line 4 (4 A1%=12345) resides at

```
6B85: 04 00 41 31 25 D5 31 32 33 34 35
```

```
BASIC 4 A 1 % = 1 2 3 4 5
```

To find where the variable A1% is stored, we need to examine the code commencing at 6D91, where we find:-

```
6D91: 02 31 41 39 30
```

```
(1) (2)- (3)-
```

(1) the Variable type - 2=integer, 4=single, 8=double, 3=string

(2) the Name in LSB/MSB format

(3) the Value in LSB/MSB format - MSB = 30H, LSB = 39H, so the HEX value = 3039H = 12345 decimal. The largest value that can be stored here is FFFFH = 65535 which is the range of an integer variable -32768 to +32767.

To find where the variable B2! is stored, if we examine the code at 6B93 we find:-

```
6B93: 05 00 42 32 21 D5 33 33 2E 31 38 37 35
```

```
BASIC 5 B 2 ! = 3 3 . 1 8 7 5
```

and the variable is stored at 6D96H as:-

```
6D96: 04 32 42 00 C0 04 85
```

```
(1) (2)- (3)---
```

(1) is the variable type - 4=Single Precision

(2) is the name in LSB/MSB format

(3) is the value stored in Floating Point Number format

Floating Point Number format is a subject all on its own. Both Single and Double Precision numbers are stored in this format, the difference being that the mantissa of a single precision variable is three bytes while the mantissa of a double precision variable is seven bytes. I will be studying the Single Precision variable.

To convert the variable B2!=33.1875 let us deal with the integer portion of the number first

```
(2[5]*1 = 32
```

```
(2[4]*0 = 0
```

```
(2[3]*0 = 0
```

```
(2[2]*0 = 0
```

```
(2[1]*0 = 0
```

```
(2[0]*1 = 1
```

33 dec. = 100001 binary

The fractional part of the number is calculated as follows

```
(1/2)[1 = .5000 * 0 = .0000
```

```
(1/2)[2 = .2500 * 0 = .0000
```

```
(1/2)[3 = .1250 * 1 = .1250
```

```
(1/2)[4 = .0625 * 1 = .0625
```

.1875 dec. = .0011 binary

The complete number therefore is 33.1875 dec. = 100001.0011 binary. This completes Step 1 in the process of converting to floating point format.

Step 2: is to convert the number to Exponential format. This is done by moving the binary point so that the entire number is a binary fraction with a "1" in the first position thus:-

```
100001.0011 --> E+6 .1000010011
```

Movement of the "point" to the left gives a positive exponent, while movement to the right gives a negative exponent. In this case we had to move the point 6 places left, so the EXPonent = 6.

Step 3: is to add 128 to the Exponent, the E+6 +128 = E+134.

Step 4: is to arrange the mantissa in bytes (three required here)

| | MSB | NMSB | LSB |
|-----------------|-----------|-----------|-----------|
| .1000010011 --> | 1000 0100 | 1100 0000 | 0000 0000 |

Step 5: is to modify the MSB to reflect the correct SIGN of the number - the lead bit is set to 0=positive: 1=negative

Since we have in the example a positive number the MSB is thus modified

```
MSB=1000 0100 --> MSB=0000 0100
```

The number may now be expressed in floating point format as

| Number | = EXP | MSB | NMSB | LSB |
|---------|---------|-----------|-----------|-----------|
| 33.1875 | = E+134 | 0000 0100 | 1100 0000 | 0000 0000 |
| | 86H | 04H | C0H | 00H |

Because of the design of the TRS-80 computer, the number is stored in Least ----> Most Significance so the computer will store the number as

| Number | = LSB | NMSB | MSB | EXP |
|---------|-------|------|-----|-----|
| 33.1875 | = 00H | C0H | 04H | 86H |

which is what is stored at locations 6D99-6D9C.

In future articles I will be looking at other aspects of variable storage using the BASIC programme in the Appendix

APPENDIX

```
1 REM=>FILENAME...VARSTORE/B...28/9/84
```

```
=>PROGRAM TO DEMONSTRATE VARIABLE STORAGE
```

```
2 REM=>ADDRESSES TO REMEMBER
```

```
40A4-40A5 (16548-16549) - START OF BASIC
```

```
40F9-40FA (16633-16634) - SIMPLE VARIABLES TABLE
```

```
40FB-40FC (16635-16636) - ARRAY VARIABLES TABLE
```

```
3 DIMA(10),A$(10),B(5,2),B$(5,2),C$(10),D$(5,2)
```

```
4 A1%=12345
```

```
5 B2!=33.1875
```

```
6 C3!=-33.1875
```

```
7 D4!=0.08125
```

```
8 D5!=-0.08125
```

```
9 FORI=1TO10:A(I)=RND(999999):NEXTI
```

```
10 X=RND(64):FORI=1TO10:A$(I)=STRING$(10,(64+X)):NEXTI
```

```
11 FORI=1TO5:FORJ=1TO2:B(I,J)=RND(999999):NEXTJ,I
```

```
12 X=RND(64):FORI=1TO2:FORJ=1TO2:B$(I,J)=STRING$(10,(64+X))
```

```

) :NEXTJ, I
13 FORI=1TO10:C%(I)=RND(32767):NEXTI
14 FORI=1TO5:FORJ=1TO2:C%(I,J)=RND(32767):NEXTJ,I
15 CLS:PRINT"SUBSCRIPT","A(I)","A$(I)","C%(I)"
16 FORI=1TO10:PRINTI,A(I),A$(I),C%(I):NEXT
17 INPUT"<CR>";Q:CLS
18 PRINT "SUBSCRIPT","B(I,J)","B$(I,J)","D%(I,J)"
19 FORI=1TO5:FORJ=1TO2:PRINTI;J,B(I,J),B$(I,J),D%(I,J):NEX
TJ,I
20 REM==>END OF PROGRAM

```

20 REM==>END OF PROGRAM

| SUBSCRIPT | A(I) | A\$(I) | C%(I) |
|-----------|-------|------------|-------|
| 1 | 16387 | GGGGGGGGGG | 29639 |
| 2 | 23169 | EEEEEEEEEE | 21933 |
| 3 | 10012 | YYYYYYYYYY | 3611 |
| 4 | 4771 | KKKKKKKKKK | 16913 |
| 5 | 12008 | UUUUUUUUUU | 28702 |
| 6 | 17627 | QQQQQQQQQQ | 25611 |
| 7 | 5508 | ----- | 12324 |
| 8 | 10169 | ZZZZZZZZZZ | 19201 |
| 9 | 20855 | TTTTTTTTTT | 3500 |
| 10 | 16092 | QQQQQQQQQQ | 15659 |

<CR>? -

Alan's Sector

by Alan Morrison (02) 625-5869

Welcome members, this is my first column published in the newsletter that you and I are so proud of. The objective of my column is to lend a few practical hints to you, and maybe some write-ups, so that you will get more from your computer. The handy hints that I will submit to you will, I hope, give you some help, and they will always be there on paper should you need them. Some of these hints you may already be aware of, but for those who aren't, well this column is for you. Unfortunately, some of these hints may not always work on your machine - this may be due to differences between various makes and models or different Operating systems etc... However, if they do not work properly, please ring me at home and I will try to help you out to the best of my ability.

Well, on with the hints. My hint for this month is how to get the new version of MICRO CHORD to do it's job properly. Some of you may have experienced difficulties in trying to save the /CMD music to disk. I stumbled across the solution by accident. I was given the fixes to create NEWCHORD from "Cornsoft's" MICRO CHORD, being a "muso", I went to work arranging music using NEWCHORD only to find, to my dissapointment, that the /CMD version of my arrangement CRASHED!

TO MAKE IT WORK ALL YOU HAVE TO DO IS....

1) Put a NEWDOS/80 System disk in drive 0 and a NEWDOS/80 Single Density data diskette in drive 1. The data disk must have the un-compiled music on it that you wish to compile.

2) Run NEWCHORD/CMD and then press "L" then "D" to get the prompt asking for the filespec. Enter the filespec and if all goes well, your music should appear on the screen.

3) Press "P" then "S" because you have to play the music before you save it to disk! This is because NEWCHORD compiles the file during the PLAY phase and will only save the music that has been previously been played.

4) When you have done that, then you are ready to to save the compiled /CMD music to disk. Pay Attention! Press "P" then "D" and you should be prompted for a filespec. Enter the filespec (make sure that you save it to drive 1) and press "ENTER".

5) Next, you will be prompted for the Start address. I use LDOS and so find the best point is at the bottom of unused memory at 5200H.

6) At this point, the disk will activate and you may get an error message - ignore the message and hit "ENTER" to get back to the main program.

7) You will notice garbage spread right across the screen - ignore it! - Follow step 4 again using exactly the same filespecs and memory addresses that you used before to ensure that the /CMD file (object code) will be saved on top of itself.

8) This time, you will notice that it took a lot less time for the program to save to disk and you should not have got any error messages. If you do get an error message, then you should try the complete sequence again using a different filename (you can always rename it later).

9) Now you are ready to make it sing (if you use LDOS you can now copy it to LDOS using MULTIDOS or COPY23/BAS) you should now have a working copy of the music.

I admit, it is a lot of mucking about but if you are as serious about music as I am, then it will be well worth the effort..... See you at Sefton - Alan.

Uploading from MICRODOS

by Brendan Thompson

Reprinted from Christchurch-80

Remember OS-80? It was also called MICRODOS. In the days before NEWDOS/80, this disk operating system from Percom used to be one of the few DOSes available in Christchurch for the TRS-80 or SYSTEM-80.

MICRODOS is a very simple DOS, little more than an enhanced Level 2 Basic with disk input/output added. In the early days a number of Dick Smith applications programs were sold on OS-80 disks and run in MICRODOS, but nowadays users want to shift them to modern DOSes.

The problem with trying to shift these programs is that OS-80 is not compatible in disk format with any other operating system. It's "system" (such as it is) is

on the first 20 sectors of the disk, and programs are dotted about the disk wherever the operator wants to put them. There is no directory as such, and the operator must remember (or note) where the programs start on the disk. Luckily the DOS does mark the end of the program, so you don't have to remember the finishing sector too.

The two main ways of transferring programs from OS-80 to your favorite DOS have traditionally been :-

1. A CSAVE to tape with later CLOAD or -
2. LLIST to printer with later entry via "opto-digital interface" (eyes and fingers).

Just about anything is better than those, so here is an "in-memory" way to transfer those useful programs from MICRODOS to a real DOS. The figures given apply to NEWDOS/80 version 2.0, but the method can be used on any other DOS as long as you know the starting address for BASIC programs on that operating system.

In the TRS-80 and SYSTEM-80, the starting address of BASIC programs is held in pointers at addresses 40A4H and 40A5H (16548 and 16549 decimal). It is possible to persuade the program to load into a different area if you alter the values in those addresses, and here lies the secret.

OS-80 loads its BASIC programs from 5A01H (23041 decimal) whereas NEWDOS/80 BASIC programs start at 6A46H (27206H). To transfer a program, all we have to do is load the OS-80 program at 6A46H and then we can boot the NEWDOS Basic under it. Here is how to do it :-

- A) "BOOT" the OS-80 disk, and <BREAK> out as soon as possible. "NEW" any program that may have been automatically loaded.
- B) "POKE" the values into the required locations. OS-80 can handle HEX with the &H prefix, so don't bother translating to decimal. The command to use is :-
POKE &H40A4,&H46 : POKE &H40A5,&H6A
- C) "LOAD" the OS-80 program WITHOUT running it. The command is LOAD xx (where xx is the Start Sector). OS-80 programs are usually called with "LOAD xx,R" which loads and then runs the program.
- D) Change disks, and BOOT NEWDOS/80. Be sure you don't have an AUTO command on your NEWDOS disk which will automatically execute some program on BOOT-UP, or you may corrupt the OS-80 program in memory.
- E) Type "BASIC *" (which will work most times) or enter BASIC and "RENEW" (which will work every time) to adjust the NEWDOS pointers around your new program. If you are using some other DOS, check the manual for the proper command (if it has one) to restore BASIC pointers in that DOS.
- F) Change the program to delete non-NEWDOS commands, and "SAVE" it to your NEWDOS disk. References to "LOAD 100,R" and the like can be changed to :-

LOAD "FILESPEC",R or RUN "FILESPEC"

where "FILESPEC" is the name you give the other OS-80 modules from your source disk.

The Comm Line

by Michael Cooper (SYDTRUG Sysop)

This month as well as the usual BBS News I have included a listing of all the PAMS (Public Access Message Systems) or BBS's that I know of in the country. You should have a lot of fun, not to speak of a lot of a bill, if you ring them all up. If you should find any numbers not working as a BBS then please let me know.

BBS News

Well callers have now exceeded 8000 and still they grow. It seems to me that there is a constant stream of new MODEM owners exploring the system, and most of them have no trouble learning to use it. The file transfer procedure leaves a few confused, but the majority seem to master it all very quickly.

I'm still working on the Trading Room software so please be patient and I'll get it up as soon as practical. In the meantime you can put all the For Sale notices in the Mail Room.

That's it for this time, see you at the meetings.

Australian & N.Z. PAMS List

Last Updated : 12th. October 1985.

This file is ordered on the basis of the cost of calling from the Sydney Telephone District.

AUSTRALIAN Systems

NEW SOUTH WALES :

SYDTRUG Club-80 RTRS (Michael Cooper)
(02) 332 2494 24 hours EST
Omen I RTRS (Ted Romer)
(02) 498 2495 1630-0900 EST Weekdays
+ V21 and V23 + 24 hours EST Weekends
AUGUR TBBS (Mark James)
(02) 661 4739 24 hours EST
Kee-Board TBBS (Phillip Keegan)
(02) 629 2230 1800-0830 EST Daily
Bresike Omen RTRS (Geoff Arthur)
(02) 457 8281 24 hours EST
Prophet TBBS (Larry Lewis)
(02) 628 7030 24 hours EST
Zeta RTRS (Nick Andrew)
(02) 627 4177 1900 to 0700 EST
Dick Smith Electronics RIBM (Steven Engel)
(02) 887 2276 24 hours EST
Texas Instruments Sydney Home Users Group
(Shane Andersen)
(02) 560 0926 1900-0700 EST Weekdays
+ MEMBERS ONLY + 24 hours Weekends

SYDTRUG NEWS - NOVEMBER 1985

| | |
|---|--|
| Mi Computer Club BBS (Rose Vines) | Newcastle Microcomputer Club RCPM (Tony Nicholson) |
| (02) 662 1686 24 hours EST | (049) 68 5383 1700-0830 EST weekdays |
| Micro Design Lab RCPM (Stephen Jolly) | 24 hours weekends |
| (02) 663 0150 1700-0700 EST Weekdays | |
| 24 hours Weekends | A.C.T.: |
| Tesseract RCPM (John Hastwell-Batten) | DSA-80 RTRS (Ross Boys) |
| (02) 651 1404 24 hours EST | (062) 41 4395 24 hours EST |
| Info-Centre BBS (Paris Radio) | Canberra RBBS (Sysop not known) |
| (02) 344 9511 24 hours EST | (061) 88 8318 24 hours EST |
| Apple Users Group BBS (Matthew Barnes & Andrew Riley) | PC Exchange RIBM (Phil Harding) |
| (02) 451 6575 24 hours EST | (062) 58 1406 24 hours EST |
| The Skull RAPL (Les Ayling) | MICSIG RCPM (Ross Elliot) |
| (02) 529 8750 24 hours EST | (062) 85 1026 24 hours EST |
| Tomorrowland DIRECT RMSD (Mike Kidson) | ACT Apple BBS (Ian Warren) |
| (02) 411 3909 24 hours EST | (062) 31 9462 2200 - 0600 EST Daily |
| Contact BBS (Steven Williams) | Treasure Trove BBS (Sysop not known) |
| (02) 550 1004 0900-2300 EST Weekdays | (062) 81 5857 24 hours |
| + MEMBERS ONLY + 24 hours Weekends | C-64 Night Time Crows Nest (Sysop not known) |
| Ausborne Users Group RCPM (Daniel Moran) | (062) 54 9821 2130 - 0730 EST Daily |
| (02) 95 5377 24 hours EST | |
| RCDM C-64 BBS (Simon Finch) | VICTORIA: |
| (02) 667 1930 24 hours EST | Omen-IV RTRS (Philip Westh) |
| + MEMBERS ONLY + * Half Duplex * | (03) 846 4034 24 hours EST |
| Commboard C-64 BBS (Graham Lee) | Melbourne PIE (Len Gould) |
| (02) 664 2334 24 hours EST | (03) 878 6847 24 hours EST |
| + MEMBERS ONLY + | C-64 BBS (Alan Miles) |
| Palantir C-64 BBS (Steve Sharp) | (03) 489 4557 24 hours EST |
| (02) 451 6576 24 hours EST | The National BBS (John Blackett-Smith) |
| Galaxy BBS (Chris Nelligan) | (03) 818 1934 1700 - 0900 EST Weekdays |
| (02) 875 3943 24 hours EST | 24 hours EST Weekends |
| Scorpio C-64 BBS (Russ Morrison) | The Dark Tower BBS (Sysop not known) |
| (02) 621 7487 24 hours EST | (03) 337 0877 24 hours EST |
| Csace BBS (Larry O'Keefe) | (Password "HACK") |
| (02) 529 8750 or 24 hours EST | Apple Hackers Uni BBS (Sysop not known) |
| (02) 529 8249 | (03) 762 1582 24 hours EST |
| A.C.E. BBS (Jeff Maddock) | (Password "AHUBBS") |
| (02) 560 9846 1800 - 0900 EST Weekdays | Basis Medfly BBS (Sysop not known) |
| 24 hours EST Weekends | (03) 500 0562 24 hours EST |
| Sydney PC Users RIBM (Geoff May) | Down Under BBS (Sysop not known) |
| (02) 238 9034 24 hours EST | (03) 429 5819 24 hours EST |
| RUNX Remote Unix System (Mark Webster) | Melbourne Micro Computer Club CBBS (Peter Jetson) |
| (02) 487 2533 (V21) 24 hours EST V21,V22 & V23 | (03) 762 5088 24 hours EST |
| (02) 48 3831 (V22) | Sorcerer Computer Users Assoc CBBS (David Woodberry) |
| (02) 487 1860 (V23) | (03) 434 3529 24 hours EST |
| Sentry BBS (Trev Roydhouse) | Tardis RCPM (Malcolm Miles) |
| (02) 428 4687 2100 - 0600 Mon - Fri | (03) 67 7760 1800-0900 EST weekdays |
| 2000 - 0600 Sat & Sun | 24 hours weekends |
| Goblin Sound RMAC (Ned Whitford) | Melbourne PC Connection RIBM (Lloyd Borrett) |
| (02) 660 8182 24 hours EST | (03) 528 3750 24 hours EST |
| + MEMBERS ONLY + | HiSoft RIBM (Richard Tolhurst) |
| Frontier Sounds RIBM (John Stanton) | (03) 799 2001 24 hours EST |
| (02) 875 2606 24 hours EST | Computers Galore RIBM (Bob Cooban and Martin Scerri) |
| BERT (Sysop not known) | (03) 561 8479 24 hours EST |
| (02) 211 0855 24 hours EST | East Ringwood RCPM (Mick Stock) |
| + V23 Videotext + | (03) 870 4623 1600 - 2400 EST weekdays |
| ABCOM RIBM (Ben Sharif) | + NOT WEEKENDS + |
| (047) 36 4165 24 hours EST | MicroPro RCPM (Sysop not known) |
| | (03) 568 8180 24 hours EST |

SYDTRUG NEWS - NOVEMBER 1985

AM-NET RCPM (Peter Hallgarten)

(03) 366 7055 24 hours EST

+ MEMBERS ONLY +

Microbee Users Group RCPM (G. Forrest)

(03) 873 5734 24 hours EST

Gippsland MAIL-BUS (Max Moore)

(051) 27 7245 24 hours EST

MIN-NET BBS (Mal Fields)

(054) 41 3013 24 hours EST

QUEENSLAND:

Software Tools RCPM (Bill Bolton)

(07) 378 9530 24 hours EST

+ V22 ONLY +

Tomorrowland DIRECT RMSD (Ian Bennett)

(07) 286 2438 24 hours EST

Texas Instruments BBS (Sysop not known)

(07) 263 6161 1900 - 0600 EST weekdays

+ NOT WEEKENDS +

Competron RIBM (Mike Walsh and David Johnson)

(07) 52 9294 24 hours EST

Hi-Tech 'C' BBS (Sysop not known)

(07) 38 6872 24 hours EST

ACEA C-64 BBS (Sysop not known)

(07) 341 0285 24 hours EST

BCUG C-64 BBS (Sysop not known)

(07) 808 2125 24 hours EST

COCO Line BBS (Kevin Mischewski)

(07) 32 6340 24 hours EST

Brisbane Microbee BBS (Sysop not known)

(07) 51 3582 1800-0800 EST Weekdays

24 hours Weekends

Cairns & District RIBM (Neil Barker)

(070) 51 3582 1800 - 0800 EST weekdays

24 hours weekends

SOUTH AUSTRALIA:

Adelaide Micro User Group RTRS (Richard Newcombe)

(08) 271 2043 24 hours CST

OMEN V RTRS (Richard Siggs)

(08) 45 4666 1800 - 0700 CST daily

Computer Ventures BBS (Daniel Schumacher)

(08) 255 9146 24 hours CST

+ MEMBERS ONLY +

The Electronic Oracle RIBM (Don Grago and Grayham Smith)

(08) 260 6686 24 hours CST

NEXUS BBS (S.A. Education Dept)

(08) 243 2477 24 hours CST

Multiple BBS (Sysop not known)

(08) 255 5116 24 hours CST

SA C-64 BBS (Sysop not known)

(08) 382 4631 24 hours CST

NORTHERN TERRITORY:

Omen-II RTRS (Terry O'Brien)

(089) 27 4454 24 hours CST

Outback RCPM (Phill Sampson)

(089) 27 7111 24 hours CST

Red Centre RCPM (Mark Little)

(089) 52 8852 24 hours CST

TASMANIA:

MS-RBBS RTRS (Mike Scott)

(003) 34 0911 24 hours EST

WESTERN AUSTRALIA:

Omen-III RTRS (Greg Watkins)

(09) 342 8555 24 hours WST

MINISYS RTRS (Mark Lillywhite)

(09) 337 2941 2100 - 1600 WST Mon - Thurs

0000 - 1600 WST Fri

0000 - 0800 Weekends

NEW ZEALAND Systems

NORTH ISLAND:

NZ Micro Club BBS (Chris Cotton)

ISD number 0011 64 9 76 2309 24 hours NZT

(type "help" to login)

Attache RBBS (Sysop not known)

ISD number 0011 64 9 78 9084 24 hours NZT

Rotorua BBS (Sysop not known)

ISD number 0011 64 73 70154 24 hours NZT

I will try to keep this listing up to date, but I cannot call all of the systems listed just to see if they are still on air. If you know of any systems not listed, or deceased ones that are, please let me know by leaving me a message as you leave the BBS. Thanks in advance, Michael Cooper.

SuperSCRIPT Patches

by P. Knaggs

Reprinted from Northern Bytes (Vol.6 Iss.5)

These are some patches that I was asked to work out by a member of our user group. I include them as they could be of some interest. Please check the Model 1/III patches though, I'm not sure if they are correct.

The standard version of SuperSCRIPT uses an underline character code on the screen as "shorthand" for every two spaces together. This has the disadvantage that the screen layout then does not match the printer layout. This is especially true of the Model 4 version (with it's 80 column screen).

To overcome this, I have located the offending code and changed it. After amendment, as described below, spaces are not compressed and the screen layout will match the printer output for all except proportional spaced printing. The two changes are to the Scriptit/ASCII conversion and the keyboard text input routines. In each case a conditional jump is replaced by an unconditional jump, removing the test for double space.

Zaps are as follows :-

MODEL I/III

SCR35/CTL FRS 30H BYTE 30H change byte 20 to 18

Context: from 2EH onwards FE 20 20 0B 2B BE

SCR64/CTL FRS 38H BYTE 38H change byte 20 to 18

Context: from 36H onwards FE 20 20 07 36 F7

MODEL 4

SCR35/CTL FRS 20 BYTE 28H change byte 20 to 18

Context: from 26H onwards FE 20 20 0B 2B BE

PATCH SCR35/CTL (ADD=99C9,FIND=20,CHG=18)

SCRIPSIT/CTL FRS 20 BYTE 30H change byte 20 to 18

Context: from 2EH onwards FE 20 20 07 36 F7

PATCH SCRIPSIT/CTL (ADD=43D4,FIND=20,CHG=18)

Repair SuperSCRIPSIT Files

from the "TRS-80 Update 1985-03"

reprinted from "Bits & Bytes" - Brisbane.

This program could be of great use to those of you running SuperSCRIPSIT on the Model 4 (and should run almost unaltered - except for screen formatting - on the Models I & III).

The following BASIC program has apparently had some success in the field with fixing blown Model 4 SuperSCRIPSIT (26-1595) documents, particularly "Record Number Out of Range" errors. This is an unofficial fix and we assume no liability for its use.

```

5 WIDTH 255 : ' <---- TESTED ON TRSDOS 6.1.2 & 6.2.0
10 ON ERROR GOTO 0 : CLS
15 PRINT STRING$(79,"*")
20 PRINT"***";SPC(73);"***"
25 PRINT"*** While reasonable efforts have been taken in
   the preparation of this ***"
30 PRINT"*** information to assure it's accuracy, TANDY
   CORPORATION assumes no ***"
35 PRINT"*** liability resulting from errors or omission
   s in this information or ***"
40 PRINT"*** from the the use of the information contain
   ed herein. ***"
45 PRINT"***";SPC(73);"***"
50 PRINT STRING$(79,"*")
55 PRINT:PRINT:PRINT SPC(25);: LINE INPUT"Press <ENTER> to
   continue...";A$
60 CLS:PRINT STRING$(79,"*")
65 PRINT"***";SPC(16);"SuperSCRIPSIT Recovery Program v.06
   .01.01";SPC(16);"***"
70 PRINT STRING$(79,"*")
75 GOSUB 3000
80 LINE INPUT"Original (unreadable) SuperSCRIPSIT filename
   .....";S$
90 LINE INPUT"Desination filename (converted to ASCII)....
   .....";D$
95 N=1
100 OPEN"R",1,S$

```

```

110 OPEN"O",2,D$
120 FIELD 1,128 AS A$, 128 AS B$
200 FOR X=7 TO LOF(1)
205 ON ERROR GOTO 2000
210 GET 1,X
220 A1$(1)="":A1$(2)=""
225 ON ERROR GOTO 0
230 AA$(1)=A$:AA$(2)=B$
240 FOR M=1 TO 2
250 FOR Y=N TO 128:K=ASC(MID$(AA$(M),Y,1))
260 IF K<32 OR K>122 THEN GOSUB 1000
270 A1$(M)=A1$(M)+CHR$(K)
280 NEXT Y: N=1 : NEXT M
300 FOR M=1 TO 2
310 PRINT A1$(M);: PRINT # 2,A1$(M);
320 NEXT M
330 NEXT X: PRINT#2," "
340 CLOSE
345 CLS:PRINT:PRINT"--- Done.. ---"
": FOR X=1 TO 3: SOUND 5,0: FOR Y=1 TO 5: NEXT Y: NEXT X
350 END
1000 IF K=247 THEN A1$(M)=A1$(M)+" ":K=32: RETURN
1010 IF K=242 THEN K=9: RETURN
1020 IF K<>253 THEN K=32: RETURN
1030 K=13:Y=Y+6:Z=Y-128: IF Z<0 THEN N=N-Z
1040 RETURN
2000 PRINT"error!!":RESUME 330
3000 PRINT:PRINT"This program will process an unreadable do
   cument (for any reason) and"
3010 PRINT"create a new file in ASCII. This new file can
   be converted back to a docu-"
3020 PRINT"ment file by selecting <A> at SuperSCRIPSIT's m
   ain menu. This recovery pro-"
3030 PRINT"gram will not alter the original file but it is
   suggested to BACKUP" the orig"
3040 PRINT"iginal file if possible before going any further
   . Also verify that enough"
3050 PRINT"disk space is available for the new file to be
   created. If the owner makes"
3060 PRINT"extensive use of BLOCKS commands, you will find
   duplicated paragraphs and"
3070 PRINT"empty sectors -
   that is normal. Also, all control codes (Underline, Bold
   , Spe-"
3080 PRINT"cial characters...) will have to be re-
   entered manually."
3090 PRINT:PRINT:LINE INPUT"                               Press
   <ENTER> to continue...";A$
3100 PRINT CHR$(28);STRING$(4,26);CHR$(31)
3110 RETURN

```

Visicalc Business Forecasting Model

reviewed by Phill Walsh

The Visicalc Business Forecasting Model (BFM) is a series of seven templates for the popular spreadsheet program. These seven templates comprise four financial

models (with three support templates) that permit business analysis and forecasting, including the projection of profit and loss, calculation of financial ratios, and balance sheet manipulation.

To use BFM, you'll need a model III/IV with 48k and at least one disk drive, Visicalc, and a printer. The BFM software presumes you have an understanding of both Visicalc and Financial modeling.

BFM is a complex financial analysis tool. It is designed primarily for manufacturing, distribution and retail business.

The BFM templates provide an Income Statement to track the performance of a business over a period of time, typically one year, a Balance Sheet, a fiscal snapshot of a company on a given date; a statement of cash flow for predicting the amount and date of needed capital; and a financial ratios statement, a common method for determining a company's performance in its marketplace.

To build these four models, one uses three support templates to calculate business' sales and costs of goods sold, salaries, and assets and depreciation.

The support templates provide data for the four models. Some of the data in each model is used by the other models as well. The sections of a model used by another model are called Drivers in the BFM language. Drivers make use of Visicalc's DIF (data interchange format) capability.

As one might suspect, the BFM models can be quite large. BFM instructions suggest a minimum of two formatted data discs for building your company's financial models.

Because of its complexity, BFM is best used after studying the manual tutorial.

The tutorial also uses several sample templates included on the master disc. Together with the written material, these samples provide a walk-through of building four financial models for the A.B.C. Model Company.

The sample templates contain most of the data for analysing A.B.C. Model's performance over a past five-year period, and the tutorial's pages provide some additional data and instructions to enter same.

The seven tutorial lessons describe how one builds the support templates, save these files, then load them into the appropriate model(s). The last lesson guides you through the complicated, labor-intensive, iterative calculation of interest rates.

The interest rate calculation requires that the program pass the information in the income statement and balance sheet templates back and forth. This is necessary because banks use different interest rates depending on an assessment of the debtor's ability to repay, usually determined by the balance sheet.

The BFM models consider interest expense to be a cost of doing business however, and include it on the income statement. Correct calculation of this expense can require as many as six exchanges between the two templates.

Each exchange involves saving the relevant figures in a DIF file, exiting the current worksheet, then loading the relevant DIF file numbers. The program makes new calculations and one repeats the process to return to the other template.

Presumably, other ways exist to calculate interest expense with less work. But for the businessman without access to financial consultants, BFM capability to make these calculations is an asset.

One can walk through the tutorial in just under two hours, and at its conclusion, most businessmen would have a good grasp of BFM's inner workings.

BFM's two principal strengths are its manual and its parent-Visicalc.

Professionally prepared by Visicalc for resale by Tandy, BFM benefits from being an application of Visicalc. One can comfortably master Visicalc in a day or so, unlike programming languages that can take years to master for complex applications.

Once you learn its command structure, you will concentrate on the important matters - your data. Most businessmen are familiar with important data about their company or department.

A Visicalc application benefits from the relatively simple modifications that you can make to a template. Modifying a BASIC or PASCAL etc program can often be a nightmare, but a spreadsheet is easy to alter to suit your own particular need.

To make BFM easy to modify, as well as use, requires a good manual. The BFM manual has three sections. Section One is a 16 page outlet describing what BFM does, its component parts, and what is needed to operate it.

Section Two is an excellent tutorial divided into seven lessons, none of which is too long. Section Three is for reference, explaining the formulas used and their rationale.

The writing is cordial, professional, and jargon-free with short paragraphs and plenty of individual headings. This makes possible the most comprehensive index I have seen in an instruction manual; six pages and over 500 cross-referenced entries.

BFM has a few weaknesses. It is complex (recall ENB?), and you can't print most of the models at once, even in 132 column format, and none of the models are directly suitable for a specific business (perhaps not so much a weakness, but merely an inconvenience since modifications are so easy).

These disadvantages are minor in the software business. Business Forecasting Model would mostly be confined to accountants and businessmen and managers of substantial business experience.

LDOS / TRSDOS Corner

by Gary Bryce (02) 628-5058

This month I will attempt to show you how to "do the impossible" with Model 1 LDOS, by creating a "Bootable" Model 1, single density, double sided SYSTEM disk. In the TRSDOS 6 section of the column I will attempt to shed a bit of light on how to operate JCL's "Lightning fast" with the aid of Memdisk.

LDOS Notes

Have you ever been told that "you can't make a Model 1, single density, double sided (SDDS) SYSTEM disk!". Wrong!!! With a little fiddling about and creating a SYS0/SYS with two directory extents (unworkable according to LDOS documentation), we can overcome the limitations imposed by the bootstrap routine which is not capable of accessing the back side of the diskette.

Before the system can access both sides of the disk, "SYS0/SYS", "SYS2/SYS" and "CONFIG/SYS" must be able to read from the front side of the disk during the "BOOTING" phase.

The following table shows the relationship between the number of sectors per granule and granules per track for single/double sided, single/double density 5K" LDOS diskettes :-

| Sides | Density | Sectors | Granules | Sect/Gran |
|-------|---------|---------|----------|-----------|
| 1 | Single | 10 | 2 | 5 |
| 1 | Double | 18 | 3 | 5 |
| 2 | Single | 20 | 4 | 6 |
| 2 | Double | 36 | 6 | 6 |

The SYS0/SYS file has 17 sectors, and therefore four grans on a single density disk and three grans on a double density disk. The point of this is that while SYS0 can be placed on the front side of a double density disk in one contiguous extent, SYS0 spans four grans and therefore two sides on a single density double sided disk. The only way to load SYS0 from a single density disk is to break it into two parts (extents), both on the front side of the disk and on contiguous tracks.

This is where the fiddling about comes in. To break the SYS0 file into two parts, the two grans on the back side of the disk are "locked out" and SYS0 is then copied to the disk.

The only other point to note before we begin the operation is that the "CONFIG/SYS" file must also be on the front side of the diskette and it's directory entry

must also appear on the front side of the directory. This is achieved by "Creating" the file "CONFIG/SYS.CCC" after fooling the system such that the destination disk appears to be single sided by resetting the "TWOSIDE" bit (bit 5) at byte "CDH" in the granule allocation table. This also means that the configuration file must not be longer than 10 sectors (if it is to remain contained on the front side).

The remaining "SYS" files are then grouped around the directory to improve the access time to them during normal operations. The JCL presented here is written to construct an eighty track single density, double sided "BOOTABLE" system disk with SYS0, SYS2 & CONFIG on the front side.

The resulting diskette can only be copied by using a "MIRROR IMAGE" backup, to maintain the relationship between SYS0, SYS2, CONFIG/SYS and the front side of the diskette.

```
%IF.          SDDSB00T/JCL
.             by Gary Bryce,
.             10 Leyte Ave,
.             Lethbridge Park,
.             N.S.W. 2770
.
. This JCL will create a single density double sided 80
. Track LDOS SYSTEM disk. The JCL is setup with Drive 0
. as the Source disk and Drive 1 as the Destination
.
. ***** NOTE *****
. It is recommended that the only active "HIGH MEMORY"
. driver be "TWOSIDE/CMD" or "PDUBL/CMD" to ensure that
. the resulting configuration file is within two grans
. in length.
.
. The programs PATCH/CMD, LOG/CMD and REPAIR/CMD must
. be resident on drive 0, before running this JCL.
.
//ALERT 1,0
//PAUSE %ID.Put a blank disk in Drive 1 & hit <ENTER>
.
FORMAT :1(Q=N,CYL=80,SDEN,SIDES=2,ABS)
.
//ALERT 1,0
//PAUSE %ID.<ENTER> if Format O.K., <BREAK> if not.
PATCH DIR/SYS.SYSTEM:1 (D00,CD=01)
REPAIR :1 (ALIEN)
PATCH SYS8/SYS.SYSTEM (D00,FF=22)
CREATE CONFIG/SYS.CCC:1 (LRL=256,REC=10)
PATCH DIR/SYS.SYSTEM:1 (D00,CD=21)
REPAIR :1 (ALIEN)
PATCH SYS8/SYS.SYSTEM (D00,FF=2A)
BACKUP SYS6/SYS:0 :1(S,Q=N)
PATCH SYS8/SYS.SYSTEM (D00,FF=23)
PATCH DIR/SYS.SYSTEM:1 (D00,23=FC)
REPAIR :1 (ALIEN)
BACKUP SYS0/SYS:0 :1(S,Q=N)
BACKUP SYS7/SYS:0 :1(S,Q=N)
```

```
BACKUP /SYS:0 :1(S,NEW,Q=N)
PATCH DIR/SYS.SYSTEM:1 (D00,23=F3)
REPAIR :1(ALIEN)
PATCH SYS8/SYS.SYSTEM (D00,FF=01)
BACKUP SYS8/SYS:0 :1(S,Q=N)
BACKUP :0 :1(NEW,U,I,Q=N)
```

```
.
. ***** NOTE *****
. To complete the process perform the following steps:-
. <1> Type LOG :0 <ENTER>
. <2> Put the newly created disk into drive 0
. <3> Type SYSTEM (SYSGEN=ON) <ENTER>
.
//EXIT
```

To re-write this JCL for a double sided 40 tracker, all patches to DIR/SYS (except to Byte CDH) and SYS8/SYS (except the last where FF=01) must have 14 Hex (20 Dec) subtracted from the values in the above JCL (NOTE* existing values are in Hex as required by the PATCH utility).

TRSDOS 6.2 JCL's

I recently updated my copy of TRSDOS 6.2.0 AN++ to TRSDOS 6.2.1 AR. After going through the process and observing the apparent "LIGHTNING SPEED" of operation of the JCL which is executed to perform the upgrade, I decided to take a look at it to work out what was going on.

The first thing I noticed was that it was in fact two separate JCL files, the first performs a number of "SYSTEM (SYSRES=" operations and enables "MEMDISK" within a standard 64K machine then copies the second JCL and PATCH/CMD to the Memdisk and initiates the second JCL in an "Execution Only" mode with the command "DO = MAKE621A/JCL:4". From thereon the second JCL operated from the Memdisk with a markedly noticable improvement in the execution speed over that obtained when a JCL is executed from a normal drive.

To perform your own JCL's in a similar manner the following points must be noted :-

- 1) The "Memdisk" JCL cannot contain any "Substitution Fields".
- 2) The "Memdisk" JCL cannot be "Compiled" as this would create a SYSTEM/JCL file on drive 0 and defeat the purpose of the whole operation (i.e. invoke the JCL with the "DO =" command).

Although you may be able to circumvent the above points by making the Memdisk drive 0 (I have not yet tried to verify this, and will let you know in the future if it is possible or not).

The JCL's used to perform the upgrade are interesting for another reason. The TRSDOS 6.2 documentation on JCL execution (Appendices A-4) note the restrictions of JCL, amongst these are stated that some commands and utilities

cannot be executed from within a JCL (i.e. BACKUP (Q=ON), BUILD, SYSGEN etc.). The BUILD command is used extensively through the MAKE621A/JCL so further experimentation seems to be in order to ascertain the true limitations of the JCL language.

For Sale

Tandy Model I

Black and White monitor, 48K RAM, Expansion Interface, Keyboard with 280B processor (system runs at 3.5 MHz), Lowercase, Gold pins, Dual drives, Joystick. Software includes Scripsit, Visicalc, Profile plus books and approx 150 games.

\$800

Contact :- Chris Walls (047) 31-2700 (B.H) or (02) 637-3114 (A.H)

Using Unloadable Disks on TANDON Drives

by Robert Jarrett

You may have found that some types of diskettes cannot be inserted into the TANDON type drive. This is the standard drive on Model 111's and 4's. I have used drives which have been made in different factories at different times so it is not just an adjustment problem and it occurs generally on disks which are slightly "fatter" such as 021floppy and is due to the leading edge fouling the drive spindle.

No amount of prodding or bending can make these diskettes load without removing the drive.

Here is a simple solution to reading one of these diskettes.

- 1) Cut a piece of paper a few millimetres narrower than the width of the diskette.
 - 2) Insert this piece of paper in the drive slot.
 - 3) Load the disk in the drive over the piece paper.
 - 4) Remove the piece of paper.
 - 5) Use the disk normally to copy off the files etc.
 - 6) By using the above technique such diskettes can be kept for use as master backup disks etc. which require infrequent use.
-