Registered by Australia Post Publication No. NBG6767.

SYDTRUG NEWS

SYDNEY TRS-80 USERS GROUP NEWSLETTER

P.O. BOX 297 PADSTOW 2211 Volume. 5 Issue. 6 February 1985

In this Issue

First off this month I will take time to thank all those who have taken the time to write articles for our newsletter over the past year, in particular those who have regularly contributed. As well, those who have done a mound of work behind the scenes without the quedos of getting their names in print, I speak here of Algis Bizys (our Membership Secretary) and Denis Pagett (our Treasurer) who organise the printing and distribution of the newsletter every month, and John Mercer who has saved me uncountable hours of two finger typing by transcribing 95% of all of the re-print material that you have seen here over the past six months or so.

TITLE	AUTHOR	PAGE
Presidential Notes	Geza Dujmovich	2
Backup COCO disks using SU+	Bill Peek	3
EDTASM Plus with the SYSTEM-80	David Sutton	4
The Comm Line	Michael Cooper	5
Orchestra 85 Reviewed	Michael Harris	7
Do the Right Thing!	John Mercer	8
Configure MODEM80 to your System	Leonard Yates	9
dBASE II Programming - Review	Larry Lewis	10
TRSDOS 1.3 - The Best DOS!	Jim Whittaker	10
Real Time Clock/Calender	Chris Tinney & Stephan Keller	12
Tandy Talks	Gary Bryce	14
Illustrated Multiplan - A Review	Peter Wignell	14
Illustrated Multiplan - A Review	Rowan Evans	15
Try this Puzzle	Leonard Yates	15
Optional NEWDOS/80 Zaps	CINTUG Newsletter	16

Meeting News

The February Special Interest Group meeting sees the continuation of Dan Lawrence's Basic Programming course at 1 PM, followed by the return of Bob Wardrop's Assembly Language programming lectures at 2 PM.

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 1st Sefton Scout Hall, 2 Waldron Rd, SEFTON. All meetings commence at 1 PM on the following dates :-

February 9th	Botany	March 9th	Monthly General Meeting
February 16th	Botany	March 16th	Special Interest Meeting
February 23th	Sefton	March 23th	South West Meeting

See you all there,
Gary Bryce, Editor

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) 1984 by SYDNEY TRS-80 USERS GROUP. All rights reserved. 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.

Presidential Notes

by Geza Dujmovich (047) 74-1685

MEETING WITH TANDY

Three members of the committee, Michael Cooper, Denis Pagett and myself, were invited late last year, by Alan Harrison of the Bankstown Computer Center, to TANDYS headquarters at Mt. Druitt to meet with Michael Lehman the National Computer Marketing Manager. The meeting is one that we have been seeking, so as to establish a close and friendly relationship with the manufacturers of our machines.

The meeting was conducted in a friendly informal atmosphere and we exchanged ideas on the mutual assistance that can be offered. We felt that both Tandy and the Club would benefit from this and the general tone of the discussion indicated that Michael Lehman and Alan Harrison were receptive and that an association will in fact become active.

One of the first visible results of the above discussions is a semi-regular column from Tandy sources advising us of various goings on such as specials and other news directly related to computer products that may or may not be big enough to make it into their advertising flyers, and hopefully with the considerably shorted lead time of our newsletter the news would be in most instances "hot off the press".

After the meeting we were shown around Tandys very impressive setup which included the sales offices, mailroom, advertising and Service department. The setting of the buildings there at Mt. Druitt is very modern and all that we saw within indicated a high degree of efficiency and friendliness.

Thanks a to all concerned and we hope to have and will work toward having an increasingly closer co-operation as a result.

CLUB SECRETARY

As you may be aware we have been working for the past couple of months without a club secretary. This has meant an increase of work for the rest of us in the committee and while things have proceeded smoothly enough it was with welcome relief that we invited Darrell Hegarty to our last meeting to check out the job, after he had indicated that he may be interested in being the Secretary. After he sat through the committee meeting we put the question of the position to Darrell. His positive answer has meant that he is now co-opted onto the committee as Secretary with all the associated committee member status. (At last somebody else to type up those b----y minutes and correspondance). Welcome to the workers Darrel hope you have plenty of spare time.

OVERSEAS MEMBERSHIP

As a result of our newsletter exchange scheme with a number of overseas TRS-80 related clubs we have had a few enqueries from people overseas wishing to be members. To these obviously dedicated computerists we can offer no other club facilities other than the Newsletter (perhaps if they wish to visit the best country then perhaps we can entertain them with our vast knowledge at club meetings) and if they have an unlimited phone account, the CBBS (Club-80). The committee decision as a result is that we welcome these people as members with the fee being set at (due mainly to the airmail Newsletter postage) $30.00 (Aust.) per year with the usual $10.00 (Aust.) joining fee. Now perhaps I can enroll my uncle to whom I am sending a Mod. I to try his journalistic trade on.

Just as a matter of interest copies of these overseas club magazines are held in the club library and now that Denis and I finally got the library cleaned out and organised they are available to members during the meetings, all you have to do is ask either one of us but as we only have one copy of each they cannot be borrowed at this time -- perhaps if we got some copied but time does not allow everything.

COLOR COMPUTER OWNERS

For some time now we have been trying to enlarge the number of Color Computer users in our group. As we are by definition a TRS-80 users group and Co-Cos are bona-fide TRS-80s we would dearly love their numbers to increase, because only with their number will support

within our group for the Co-Co grow. Gary Bryce and Michael Cooper addressed a meeting of the Bankstown Color Computer users at the computer center on Saturday the 12th. of January. They explained what we are all about and that we can offer the existing facilities of the meeting premises, Newsletter and Bulletin board as well as the expertise of our members many of whom own and have knowledge on a number of different TRS-80 type systems. Michael also gave a demonstration of the use of the BBS and by all reports the attending Co-Co owners were impressed by what they saw and heard. As a direct result of these efforts we got three people to come and attend our meeting that afternoon and subsequently to join our club. We're glad to have you fellows and we hope that with the continued increase in membership with people of similar interest our club can offer your systems more and more support- after all the number of Co-Cos sold must rival all of the other machines we support put together.

MEETING SHYNESS

It has been brought to my attention by a letter from a member that there may be some people attending our meetings who are not getting out of it all as much as they have hoped for. The primary reason for this as I see it stems from the nature of the meetings as well as the people having this concern. We as a club can but provide the facilities. However it is still up to the individual to get up and seek out those he sees at the meeting with obvious similar problems and interests, and although at each Monthly General meeting I ask for discussion, ideas or comments from the members the bottom line is that we cannot answer or cater for unasked questions or problems.

There is absolutely no need for the new user to re-invent the wheel, there will most certainly be someone there that has been there and done that. Most of these members are only too pleased to help those who are new or are having some interesting and unusual problems with the use of their systems. But you must speak up and/or seek out. Remember I think it's written somewhere "God helps those that help themselves". If you attend one of our - in my opinion reasonably friendly and informal meetings - but choose to sit in a corner and be shy it is quite possible that the meeting will not answer your particular computer related questions.

I hope the above is read in the spirit in which it was meant and that we see you all sooner or later at one of the meetings going around and sharing each others knowledge.

Just finally the last remaining edition of the quotations from the famed Edsel Murphy and his laws of the behaviour of inanimate objects.

* In computer hardware or software an item characterised by a number of plus or minus errors, the total error will be the sum of all errors adding in the same direction.

The source material listing references to the laws quoted over the last few issues of this newsletter was misplaced according to yet another of these laws. In accordance with these laws these misplaced documents will turn up on the date that they are no longer needed.

The man who developed one of the most profound concepts of the twentieth century is practically unknown to most people. He is a victim of his own law. destined to a secure place in the computing hall of fame, something went wrong. His real contribution lay not merely in the discovery of the law but more in its universality and in its impact. The law itself though inherently simple, has formed a foundation on which future generations will build an ever colapsing result. In fact the law first came to him in all its simplicity when his bride-to-be informed him of the impending birth of an heir to the family misfortunes.

Until next issue ---- All the best in computing .....Geza

Backup a COCO Disk using SUPER UTILITY PLUS

by Bill Peek of the Indiana Software Group
Reprinted from "NORTHERN BYTES", VOL. 5, NO. 6

This one's short and sweet - use drive configuration settings T30,35,17 and Super Utility will be able to back up your Color Computer disks on your Model I/III/4.

EDTASM PLUS Disk ver 1.01 with SYSTEM 80's

by David Sutton

If using Microsoft's EDITOR/ASSEMBLER-PLUS Version 1.01 under NEWDOS, you would be aware that the internal cassette on the System-80 is not activated by EDTASM PLUS cassette commands. This is due to different port usage on Model I & System-80 cassettes.

There are two ways of fixing this depending upon whether or not you know the start, end, and entry addresses of your copy of EDTASM PLUS. My version starts at 5780H, and ends at 900DH, and entry is at 5780H.

Method 1 - if address known.

Execute EDTASM and type in the following with Org at 4 bytes less than EDTASM start address. In my case :-

```
0010        ORG     577CH
0020        LD      A,0
0030        OUT     (254),A
0040        END
```

Asssemble to DISK as PATCH1
Return to DOS
Use DOS commands:-
LOAD PATCH1/CMD
LOAD EDTASMPL/CMD
DUMP THISCOPY/CMD:drive,577CH,900DH,577CH

THISCOPY/CMD now contains the short listing above with EDTASMPL appended with start 577CH, end 900DH, and execution 577CH. For different start addresses, amend ORG in line 10 & addresses in the DUMP command.

After executing THISCOPY/CMD all cassette commands should now activate the internal cassette.

METHOD 2 - If address unknown

The listing below should give the desired result:-

```
0010        ORG     0F000H     ;ADJUST AS REQUIRED
0020 START  LD      HL,MES1
0030        CALL    01C9H
0040        CALL    4467H
0050        LD      A,0
0060        OUT     (254),A
0070        LD      HL,MES2
0080        CALL    4405H
0090 MES2   DEFM    'EDTASMPL/CMD'     ;ADJUST AS REQUIRED
0100        DEFB    0DH
0110 MES1   DEFM    'EDITOR ASSEMBLER NOW LOADING'
0120        DEFB    0DH
0130        END     START
```

To use:- Type in the listing & adjust lines 10 & 90 as required. Note line 90 "MUST" reflect the correct filespec for your copy of EDTASM PLUS.

Assemble to disk as PATCH2

When wishing to use EDTASM PLUS execute PATCH2/CMD. The screen should clear, give a short message, and execute EDTASM PLUS. All cassette commands should now activate the internal cassette.

For Sale

Aerocomp "DDC" Doubler, near new - $120 - Lindsay Douglas (043) 25-2254 (evenings).

The Comm Line

by Michael Cooper (Club-80 SYSOP)

Well here we are into 1985, and still so many things to get done. This time I will show you how you can use an LDOS Filter Program to protect your system against tampering by callers.

CONNECTING UP TO A BBS

I know many of you are a bit apprehensive when first using a MODEM, but there is nothing to fear as the procedure is simple. First you load a terminal program in your Computer, to which is connected a MODEM. With the MODEM switched to "Phone" mode you dial the BBS number. When the system answers you will hear a tone. At this point you switch the MODEM to the "Modem" mode and thats it. Once the BBS detects your Carrier it will begin to send characters and you then have to follow the prompts and type your response.

CORRECTION

Last months program to Auto-Answer was missing a line, please insert it as :-

```
135 CLEAR 1000:DEFINT A-Z:TRUE = NOT FALSE
```

CLUB-80 BBS News

Business was brisk over the festive season, I guess there were a few MODEMS to found in the Christmas stockings around the country. One thing I find quite facinating is the number of long distance callers, some spend 30 - 40 minutes at peak STD rates to use the system.

On Saturday 12th. January I had the pleasure to join Gary Bryce at the Bankstown Computer Centre, where a meeting of local CoCo Users was in progress. Alan Harrison the manager, had invited us to attend and address the gathering. I had the opportunity to demonstrate CLUB-80 using a CoCo running the COMPAC Terminal program. It was quite an impressive little program, cassette based and with several neat features including a BASIC and M/L transfer feature. This would allow 2 CoCo owners to send and receive programs over a MODEM link.

Current callers to the BBS stands at 3300. The Discussion Room and Library are the most heavily used areas. Geoff Niland's Group for Adventure Gamers is very popular with Visitors to the BBS. It seems that it clearly fills a need to allow Adventurer's to find the answers to that which perplexes them. I have also started a new group for CoCo Users called "The CoCo Cafe". I hope this will encourage members with CoCo machines to start using CLUB-80 and providing some added input. Anyone who knows of a Christensen Transfer Program to run on a CoCo or at least a 6809 based system, please contact me.

PROTECTING YOUR SYSTEM

Since the CLUB-80 software was to run in BASIC under LDOS I needed to devise a way of stopping callers from getting into the system it crashed. The simplest way to do this is by testing that the 2 bytes at 40A2H and 40A3H both contain FFH (255). Once BASIC stops running these bytes will change, so you just have to monitor them. A couple of ways suggest themselves, you could add a routine to the interrupt chain or if you use LDOS you can add a FILTER. The latter method was the one I chose (simpler) and it works on the *KI Device Driver. When you drop to BASIC READY the ROM calls the keyboard input routine and the FILTER activates a Reboot. The FILTER may be turned ON/OFF by changing a nominated byte in memory. To use it just assemble the code as "READY/FLT" and then type at LDOS READY :

```
FILTER *KI READY/FLT
```

You will get a error message if anything is wrong, otherwise the program will relocate itself under the HIGH$ pointer and reset HIGH$ to protect itself. To switch it ON or OFF from within BASIC is done using POKE Memory,0 or 1. It can be SYSTEM (SYSGEN) 'ed into the CONFIG file.

The source code is in the CLUB-80 Library if you are a member, otherwise happy typing. If you need any more information, either leave me a message or see me at a SYDTRUG meeting. Good luck.

```
                00100 ; READY/FLT     Revision 1.0.0  20/06/84
0000            00110         TITLE   (READY/FLT : CLUB-80 FAILSAFE FILTER)
                00120 ;****
                00130 ;       READY/FLT is the Failsafe for CLUB-80 CBBS
                00140 ;       Intercepts input at command level (BASIC or DOS)
                00150 ;       and reboots the system software, denying access.
                00160 ;       Filter switched ON or OFF via byte at SWITCH
                00170 ;       BASIC program should POKE SWITCH,1 = ON
                00180 ;                              POKE SWITCH,0 = OFF
                00190 ; TRS-80 TeleComputing Software
                00200 ; (C)opyright Michael Cooper  June 1984
                00210 ;****
000A            00220 LF      EQU     10
000D            00230 CR      EQU     13
0060            00240 @PAUSE  EQU     60H
402D            00250 @EXIT   EQU     402DH
4030            00260 @ABORT  EQU     4030H
4467            00270 @DSPLY  EQU     4467H
FFFC            00280 SWITCH  EQU     0FFFCH          ;Free byte (change to suit)
                00290 ;****
                00300 ;       Model I equates as comments
                00310 ;       Change if using a Model I
                00320 ;****
4411            00330 HIGH$   EQU     4411H           ;4049H for Mod 1
428A            00340 @LOGOT  EQU     428AH           ;447BH for Mod 1
4299            00350 @CMDI   EQU     4299H           ;4405H for Mod 1
                00360 ;
5200            00370         ORG     5200H
                00380 ;
5200 1A         00390 ENTRY   LD      A,(DE)          ;Get DCB Type Byte
5201 CB5F       00400         BIT     3,A             ;Device set to NIL ?
5203 204F       00410         JR      NZ,DEVNIL       ;Go if nil
5205 CB67       00420         BIT     4,A             ;Device routed ?
5207 2054       00430         JR      NZ,DEVROUT      ;Go if routed
5209 CB47       00440         BIT     0,A             ;Device capable of input ?
520B 283E       00450         JR      Z,NOTINP        ;Go if no input
520D D5         00460         PUSH    DE              ;Save DCB
520E 216652     00470         LD      HL,HELLO
5211 CD6744     00480         CALL    @DSPLY          ;Show sign-on message
5214 DDE1       00490         POP     IX              ;IX => DCB of Device
5216 2A1144     00500         LD      HL,(HIGH$)      ;Now ready to move the
5219 012600     00510         LD      BC,LAST-START   ; filter into high memory
521C AF         00520         XOR     A
521D ED42       00530         SBC     HL,BC
521F 221144     00540         LD      (HIGH$),HL      ;Reset the High pointer
5222 23         00550         INC     HL
5223 DD7E01     00560         LD      A,(IX+1)        ;Old Device entry point
5226 32C852     00570         LD      (PUTBYT+1),A    ;Store in our filter
5229 32E952     00580         LD      (GETBYT+1),A
522C DD7E02     00590         LD      A,(IX+2)        ;Also old MSB of Device
522F 32C952     00600         LD      (PUTBYT+2),A
5232 32EA52     00610         LD      (GETBYT+2),A
5235 F3         00620         DI
5236 DD7501     00630         LD      (IX+1),L        ;Put new entry point
5239 DD7402     00640         LD      (IX+2),H        ; into the Device DCB
523C EB         00650         EX      DE,HL
523D 21C552     00660         LD      HL,START
5240 EDB0       00670         LDIR                    ;Move the filter up
5242 FB         00680         EI
5243 3E00       00690         LD      A,0             ;Init Switch to OFF
5245 32FCFF     00700         LD      (SWITCH),A      ;
5248 C32D40     00710         JP      @EXIT           ;All done
                00720 ;
524B 218652     00730 NOTINP  LD      HL,OUTMSG       ;Not input device error
```

```
524E CD8A42   00740          CALL    @LOGOT
5251 C33040   00750          JP      @ABORT
              00760 ;
5254 21A252   00770 DEVNIL   LD      HL,NILMSG       ;Device not active error
5257 CD8A42   00780          CALL    @LOGOT
525A C33040   00790          JP      @ABORT
              00800 ;
525D 21B452   00810 DEVROUT  LD      HL,ROUTMSG      ;Device is routed error
5260 CD8A42   00820          CALL    @LOGOT
5263 C33040   00830          JP      @ABORT
              00840 ;
5266 52       00850 HELLO    DEFM    'READY : CLUB-80 Failsafe Filter',CR
45 41 44 59 20 3A 20 43
4C 55 42 2D 38 30 20 46
61 69 6C 73 61 66 65 20
46 69 6C 74 65 72 0D
5286 52       00860 OUTMSG   DEFM    'READY filter for input only',CR
45 41 44 59 20 66 69 6C
74 65 72 20 66 6F 72 20
69 6E 70 75 74 20 6F 6E
6C 79 0D
52A2 44       00870 NILMSG   DEFM    'Device not active',CR
65 76 69 63 65 20 6E 6F
74 20 61 63 74 69 76 65
0D
52B4 44       00880 ROUTMSG DEFM    'Device is routed',CR
65 76 69 63 65 20 69 73
20 72 6F 75 74 65 64 0D
              00890 ;****
              00900 ;        Actual filter code to move to  High memory
              00910 ;****
52C5 3803     00920 START    JR      C,OK            ;If carry, means input request
52C7 C30000   00930 PUTBYT   JP      0               ;Old Driver for output
52CA 3AFCFF   00940 OK       LD      A,(SWITCH)      ;Failsafe enable byte
52CD FE01     00950          CP      1               ;Test it
52CF 2017     00960          JR      NZ,GETBYT       ;Go if Filter OFF
52D1 3AA240   00970          LD      A,(40A2H)       ;BASIC Current Line Number LSB
52D4 FEFF     00980          CP      0FFH            ;
52D6 2010     00990          JR      NZ,GETBYT       ;Not at READY
52D8 3AA340   01000          LD      A,(40A3H)       ;BASIC Current Line Number MSB
52DB FEFF     01010          CP      0FFH            ;
52DD 2009     01020          JR      NZ,GETBYT       ;Not at READY
52DF 010080   01030          LD      BC,8000H        ;Delay count
52E2 CD6000   01040          CALL    @PAUSE          ;Do it
52E5 C30000   01050          JP      0H              ;Boot
52E8 C30000   01060 GETBYT   JP      0               ;Old Driver for Device input
52EB          01070 LAST     EQU     $
5200          01080          END     ENTRY
```

Music for the Model I using ORCHESTRA-85

by Michael J. Harris

Hi this is my first article for a long while. In this column I will talk about ORCHESTRA-85 for the model I. Tandy sells a version called ORCHESTRA-90 for the Model III/4. All my comments also apply to the Model III version.

ORCHESTRA-85 comes in a plastic bag with a 45 page manual, a tape containing the tape version on one side and the disk version on the other side and a small circuit board. The board is about 8 cm by 6 cm and contains on the Model I version 7 I.C's , 2 resistor packs and a cap.

To install you plug the 40 pin connector in to the 40 pin bus expansion port on the left side of the E.I. Two RCA sockets are used for the connections to the AUX sockets of your stereo radio.

After loading in the software the system ask you if you have a Clock speed up kit installed, is it software controlled and what speed is it (2.66,3.54,4.00). It then asks you if you want to make a copy of this configured version, so you can just load and run it. After configuring the software is ready to run, To load a file you enter READ (or R) filename. Entering SCORE to compile the music and PLAY to run the compiled source. Some of the other command are DIR KILL APPEND LIST and GET which loads several files to play one after the other. One command which is only in the DISK version is MULTI which is a perpetual GET command.

The EDITOR is full-screen and has a search command, the editor is smooth and fast. The synthesizer has a six octave range and can produce up to five simultaneous note or voices in two stereo channels. Each voice may be assigned to any of five user-defined tone color registers. The default tone colour registers mimic the sound of a trumpet, oboe, clarinet, violin and pipe organ. The registers are easily altered to imitate other instruments, even percussion, or to make strange new sounds. I have used the system for 3 months and found it easy and fast to use.

DO THE RIGHT THING!
or Don't Litter the String Space with Unnecessary Garbage

by John Mercer.

Almost everyone who has had occasion to use a BASIC programme which does much string manipulation will have experienced the frustration of the Garbage Collection Run! For the benefit of those of you who are not familiar with this phenomenon, when the programme modifies an existing string variable it (usually) uses up some more of the space allocated for strings, and changes the pointer to reflect the new location of the string. When the programme does a lot of this, it is not long before the available space is used up. Then BASIC will temporarily suspend application programme running while it goes through the string space and relocates things. This is commonly referred to as the Garbage Collection Run.

It can be eliminated or at least reduced in frequency by a little care in programming. For those who are using DISK BASIC there are two or three string manipulation statements which will take effect in the existing location in the string space, and not requisition more space.

These are LSET, RSET and MID$( )=.

The following example may give you some idea of what I mean. One of the programmes which I have in my collection is one which handles personal accounts. It is organised in the form of fixed length records, without any spaces between the fields in order to conserve memory and disk space. When ever the records are printed, or displayed on the screen they would be quite unintelligible unless the fields were separated. The records are of such a length as permits them to be displayed on one line of the screen, and in the Model 4 version which I use, the display shows a page of twenty records at a time. When you have studied the records displayed you can press any key and the records are cleared from the screen and another page is displayed.

Originally the portion of the programme which displayed the records was something like this:

```
2010 FOR I=1 TO W
2020  GOSUB 5000: GOSUB 5020: PRINT VX$: IF P THEN LPRINT VX$
2030  L=L+1: IF L=20 THEN L=1: PRINT@1840,S0$;: GOSUB 40500 : PRINT@320,CHR$(31);
2040 NEXT: PRINT@1840,"End of Data - "S0$;: GOSUB 40500: GOTO 1002
 .
 .
5000 V1$=LEFT$(A$(I),2): V2$=MID$(A$(I),3,2): V3$=MID$(A$(I),5,2): V4$=MID$(A$(I),7,6):
     V5$=MID$(A$(I),13,31): V6$=MID$(A$(I),42,2): V7$=MID$(A$(I),44,3): V8$=MID$(A$(I),47,8):
     V9$=MID$(A$(I),55,8): RETURN
 .
5020 VX$=V1$+"/"+V2$+"/"+V3$+" "+V4$+"  "+V5$+"  "+V6$+V7$+" "+V8$+"   "+V9$: RETURN
 .
 .
40500 A$=INKEY$: IF A$=""THEN 40500 ELSE RETURN
```

When this portion of the programme was running the usual procedure was that about fifteen records would be displayed, then the shop steward would call "Everybody Out!", and there would be a break for tea, or whatever. After the "Stop Work Meeting" things would start up again. As there were sometimes six or seven video pages of records to be displayed it got somewhat frustrating. So I decided to do something about it. The altered coding is shown below:

```
2010 DEF FNM$(A,A1)=MID$(A$(I),A,A1): VX$="  / /"+STRING$(72,32)
2020 L=1: FOR I=1 TO W
2030 MID$(VX$,1)=FNM$(1,2): MID$(VX$,4)=FNM$(3,2): MID$(VX$,7)=FNM$(5,2 )
2040 MID$(VX$,10)=FNM$(7,6): MID$(VX$,18)=FNM$(13,31): MID$(VX$,51)=FNM $(44,2)
2050 MID$(VX$,53)=FNM$(46,3): MID$(VX$,59)=FNM$(49,8): MID$(VX$,71)=FNM $(57,8)
2060  PRINT VX$: IF P THEN LPRINT VX$
2070  L=L+1: IF L=20 THEN L=1: PRINT1840,S0$;: GOSUB 40500: PRINT@320,CHR$(31);
2080 NEXT: PRINT@1840,"End of Data - "S0$;: GOSUB 40500: GOTO 1002
```

When the programme was run in this form it would display seven video pages of data without any interruption.

Some comments may be in order. Line 2010 defines a User Function and establishes the String Variable VX$ with the slashes for the date and ensures that it is the correct length. That should be the last time that it will take up any String Space. By the way the User Function is in my opinion a very under-utilised function of BASIC, and one which deserves to be used more often. This particular example saves six bytes of code every time it is invoked, and you don't have to remember the exact format of the full expression which it is replacing every time you use it. When the original version was running, every time through the loop the programme requisitioned space for nine new intermediate string variables and no less than 78 bytes for the new edition of VX$. So its not hard to see how you can quickly use up string space. The programme itself is quite long and the number of data record which can be accomodated is dependent upon how much memory is available. S0$ is a message "Press any key to continue". Line 40500 is taken from Lewis Rosenfelder's "Basic Faster and Better", which I consider should be required reading for all BASIC programmers. Line 1002 which is refered to at the end is the Main Menu of the programme.

The next time you have to wait while the Garbage Collector does its thing why not look at the possibility of modifying the programme. You may be surprised by the results, (pleasantly, I hope).

Configure MODEM80 to suit your System

by Leonard Yates

(a) Hard-Configure GRAPHICS ON: Since the Club-80 BBS now has graphics capabilities, you may like to zap your copy of MODEM80 as follows so that it initializes in the GRAPHICS ON mode:

MODEM80/CMD,05,B1 change E6 7F 21 to E6 FF 21
MODEM80/CMD,28,CB change 4F 46 46 0A to 4F 4E 20 0A

(Don't forget to reset GRAPHICS OFF if accessing a remote system which checks parity.)

(b) Configuration to Suit the AVTEK Multi-Modem: To set high the DTR and RTS signals of the RS-232C serial port; you cannot use MODEM80 with an AVTEK Modem unless you make the following changes:

MODEM80/CMD,04,3D change 55 6D 3E to 55 6C 3E
MODEM80/CMD,35,D6 change 3E 6D D3 to 3E 6C D3

(c) Configuration for NEWDOS/80 version 2.0 / Model III: In its original form MODEM80 does not allow use of the option '(C) = Accept DOS Command' when using NEWDOS/80 version 2.0 on a Model III. A way of accessing the DIR command is given on page 32 of the MODEM80 manual; the following method allows all DOS commands to be accessed:

MODEM80/CMD,12,58H change 3D 28 09 to 3D 20 09

Note: If you're modifying MODEM48/CMD or a DUMPed version of MODEM80, the relative byte in each ot the above zaps should be decreased by 8. For example, byte B1 would become A9, etc.

dBasell Programming - by Peabody & Seabrook

a reveiw by Larry Lewis

The publication is intended to be for business people embarking on writing their computer programs using dBasell.

About one third of the book deals with the theory of analysis and programming. Although a good introduction to the subject the text was very evangelistic which could frighten off novices, also the methodology for analysing and programming is the subject of many other books. I don't think this one is the place to learn how to analyse or learn logical programming skills.

The remainder of the book deals with dBasell itself and as a beginners book, is very good, but then the manual provided with the package is also. A number of well put together examples of how to use the programming features are shown in an easy to understand format.

Decision Logic, usually avoided in beginners books, is tackled and explained quite satisfactorily. If studied closely the various options of branching and logic control can be mastered.

Overall, for newcomers, the book at $25-50 (published by Prentice-Hall of Australia Pty. Ltd.) is a good buy.

Model III TRSDOS 1.3 - The Best DOS !

by Jim Whittaker

A lot of people often talk about such and such a DOS being the best DOS and how they use one DOS exclusively. Well I am familiar with 3 DOS's and I use each one regularly and in about the same proportion. As far as I am concerned, its horses for courses and some DOS's are great for doing work in some areas and lousy in others and if you don't exploit their benefits, a lot of potential power is going to waste.

NEWDOS/80 I use if I need to do any file manipulating or transferring. It has the best range of options for reading files, manipulating data and then writing either to the same file or a new file in different format.

LDOS I use when I need to handle a special file or print job where I can quickly and easily install a filter or link to another device (real or imaginary). The best area to demonstrate this is when MODEMING where the type ahead buffer and easy file handling allow hassle free operations.

TRSDOS I use with most of my standard Radio Shack programs. Cobol and Superscripsit are two large systems that are split into many smaller files for manageability. TRSDOS, being a "bare bones" DOS, is not very large and can thus accommodate all these programs and many of my own files on the one system disk. This allows "1 DISK" operation which I find is more convenient to use. WIFE/CMD also finds it is less hassle to find and boot a single disk.

I find under the other DOS's that although powerful, you pay for it with lack of user space on the system disk. Often, people transfer these programs to LDOS or NEWDOS, but there is no real advantage as you cannot use type ahead or filters or other enhancements and there is very little room left on the system disk for your own files. The R/S programs use their own drivers and do NOT refer to the DOS routines. The other problem is that although the programs don't use the DOS, the DOS still thinks that it does and consequently everything runs a lot slower trying to interpret interrupts that are not used. Cobol and Superscripsit and the like were written specifically to run under TRSDOS and this is where they perform at their best.

To make life just a little easier, you will find below a whole heap of TRSDOS 1.3 system patches and enhancements that have been collected from here there and everywhere over the years. My compliments to the original authors (unknown) for their perseverance and foresight. You can either enter them individually from DOS Ready or put them in a DO file. I will upload them to the BBS as a DO file called TRSPATCH/BLD and they will be available at the club. You can change any of them by putting the file through Superscripsit's Ascii option, making the changes, deletions or additions, QUIT the document and reuse the ASCII option.

MAJOR PATCHES TO THE TRS-DOS SYSTEMS.

1. Disable Password checking in both PATCH and DEBUG.

```
PATCH *2 ( ADD=4ED4, FIND=20, CHG=18 )
PATCH *5 ( ADD=52EB, FIND=CB, CHG=36 )
PATCH *5 ( ADD=52ED, FIND=BE, CHG=00 )
```

2. Allow DEBUG to inspect below 5600H.

```
PATCH *5 ( ADD=4EDF, FIND=38E5, CHG=0000 )
PATCH *5 ( ADD=4F04, FIND=D0, CHG=C9 )
PATCH *5 ( ADD=506E, FIND=38E3, CHG=0000 )
```

3. Correct BASIC arrow functions (one or both may already be done).

```
PATCH BASIC/CMD ( ADD=58C4, FIND=D5, CHG=00 )
PATCH BASIC/CMD ( ADD=58F8, FIND=F1, CHG=00 )
```

4. Start directory searches from sector 3 not 2 on track 17.

```
PATCH *10 ( ADD=4E47, FIND=02, CHG=03 )
PATCH *10 ( ADD=4E2A, FIND=3ADA4E, CHG=784F00 )
```

5. Only zero the first two bytes in the directory when KILLing a file. This allows you to UNKILL them if need be.

```
PATCH *3 ( ADD=4FB5, FIND=2F, CHG=01 )
```

6. Correct an error in FORMAT an I/O retry counter error.

```
PATCH *7 ( ADD=4E55, FIND=12, CHG=0F )
PATCH *7 ( ADD=4E5B, FIND=0C, CHG=09 )
```

7.(a) Disable boot-up Logo.

```
PATCH *0 ( ADD=4E85, FIND=216C51, CHG=C39D4E )
```

(b) Disable both Date and Time prompts.

```
PATCH *0 ( ADD=4EA9, FIND=CA, CHG=C3 )
```

or selectively eliminate the boot up logo.

8.(a) Eliminate Model III picture.

```
PATCH *0 ( ADD=4E89, FIND=1B, CHG=27 )
```

(b) Eliminate message from "VERSION to SERIAL NUMBER".

```
PATCH *0 ( ADD=4E95, FIND=1B, CHG=27 )
```

(c) Eliminate TANDY copyrite message.

```
PATCH *0 ( ADD=4E9B, FIND=1B, CHG=27 )
```

(d) Disable Time prompt only.

```
PATCH *0 ( ADD=4EFE, FIND=215451, CHG=C32E4F )
```

9. Print long error messages instead of numbers.

```
PATCH *4 ( ADD=4E28, FIND=20, CHG=18 )
```

10.Remove periods from DOS READY prompt.

```
PATCH *1 ( ADD=4E78, FIND=2E, CHG=20 )
```

11.Change TRSDOS READY prompt (max 12 characters). Because of the length of this patch, I have split it into 2 lots of 6 characters. Just enter the new message in HEX format. NB a space = 20 HEX.

```
PATCH *1 (ADD=509C, FIND=545253444F53, CHG=535550455220)
PATCH *1 (ADD=50A2, FIND=205265616479, CHG=204D4943524F)
```

12.Change Track Step rate. Use the table below to select the different rates of step and use them in the "CHG=" spots.

Current	6ms	12ms	20ms	30ms
0C	08	09	0A	0B
1C	18	19	1A	1B
58	58	59	5A	5B

```
PATCH *0 ( ADD=42EE, FIND=0C, CHG=08 )
PATCH *0 ( ADD=4516, FIND=0C, CHG=08 )
PATCH *0 ( ADD=4544, FIND=1C, CHG=18 )
PATCH *0 ( ADD=4FE1, FIND=0C, CHG=08 )
PATCH *7 ( ADD=580E, FIND=0C, CHG=08 )
PATCH *7 ( ADD=5841, FIND=0C, CHG=08 )
PATCH *7 ( ADD=5923, FIND=1C, CHG=18 )
PATCH *7 ( ADD=5B3C, FIND=58, CHG=58 )
```

13. Change Track Count. Mainly (but not limited to) to use 80 track drives. Select the count (in HEX) and use it in the "CHG=" spots.

Current	40T	42T	80T
	28	2A	50
	29	2B	51

```
PATCH *0 ( ADD=4926, FIND=28, CHG=2A )
PATCH *0 ( ADD=499B, FIND=28, CHG=2A )
PATCH *0 ( ADD=4B29, FIND=29, CHG=2B )
PATCH *2 ( ADD=4F61, FIND=28, CHG=2A )
PATCH *6 ( ADD=5C06, FIND=28, CHG=2A )
PATCH *6 ( ADD=5D53, FIND=28, CHG=2A )
PATCH *7 ( ADD=5203, FIND=28, CHG=2A )
PATCH *7 ( ADD=52D4, FIND=28, CHG=2A )
PATCH *7 ( ADD=53FF, FIND=28, CHG=2A )
PATCH *7 ( ADD=54D7, FIND=29, CHG=2B )
PATCH *7 ( ADD=5504, FIND=28, CHG=2A )
PATCH *7 ( ADD=554A, FIND=28, CHG=2A )
PATCH *7 ( ADD=5C4F, FIND=28, CHG=2A )
PATCH *7 ( ADD=5CD4, FIND=28, CHG=2A )
```

Now for some welcome additions to TRS-DOS.

14.This addition will halve the loading time of BASIC programmes.

```
PATCH BASIC/CMD (ADD=5BFE, FIND=2AA440, CHG=CD8754)
PATCH BASIC/CMD (ADD=5C07, FIND=FF, CHG=FE)
PATCH BASIC/CMD (ADD=5C0D, FIND=CD535F7723, CHG=CD3B540000)
PATCH BASIC/CMD (ADD=53CC, FIND=8754, CHG=4A1E)
PATCH BASIC/CMD (ADD=5487, FIND=E17EFE26, CHG=2323E5DD)
PATCH BASIC/CMD (ADD=548B, FIND=C24A1ED7, CHG=E1ED5BA4)
PATCH BASIC/CMD (ADD=548F, FIND=C24A1EE5, CHG=40013300)
PATCH BASIC/CMD (ADD=5493, FIND=219354CD, CHG=0901FF00)
PATCH BASIC/CMD (ADD=5497, FIND=3F56E1C9, CHG=EDB0EBC9)
PATCH BASIC/CMD (ADD=549B, FIND=35013D3C, CHG=DD7503DD)
PATCH BASIC/CMD (ADD=549F, FIND=26751734, CHG=7404DD36)
PATCH BASIC/CMD (ADD=54A3, FIND=2633C3675, CHG=0500DDCB)
PATCH BASIC/CMD (ADD=54A7, FIND=3C2S7516, CHG=01EEDD34)
PATCH BASIC/CMD (ADD=54AB, FIND=1A050C07, CHG=0A2003DD)
PATCH BASIC/CMD (ADD=54AF, FIND=1C121D01, CHG=340224C3)
PATCH BASIC/CMD (ADD=54B3, FIND=1011, CHG=535F)
```

15. Create a "R"epeat command similar to NEWDOS80.

```
PATCH *1 (ADD=4E36, FIND=2S42013F003600, CHG=8041013F000000)
PATCH *1 (ADD=5191, FIND=4D4153544552, CHG=522020202020)
PATCH *9 (ADD=603E, FIND=CD8E5FFE0D2003, CHG=218041112542 01)
PATCH *9 (ADD=6045, FIND=AF18291110S0CD, CHG=2500EDB02125 42)
PATCH *9 (ADD=604C, FIND=5444C2, CHG=C39942)
```

16.Change the CONVERT command to allow Y/N/Q selection.

```
PATCH CONVERT/CMD (ADD=520A, FIND=06, CHG=1C)
PATCH CONVERT/CMD (ADD=5511, FIND=21955B, CHG=CD065A)
PATCH CONVERT/CMD (ADD=5536, FIND=ED53, CHG=135A)
PATCH CONVERT/CMD (ADD=5A06, FIND=4D6F646556C20,
      CHG=3E86CD155A21)
PATCH CONVERT/CMD (ADD=5A0C, FIND=3120746F204D,
      CHG=955BC0E1C39E)
PATCH CONVERT/CMD (ADD=5A12, FIND=6F646556C2033,
      CHG=553E7032F153)
PATCH CONVERT/CMD (ADD=5A18, FIND=20438F6E7865,
      CHG=CDED53C9436F)
PATCH CONVERT/CMD (ADD=5A1E, FIND=7373696F6E,
      CHG=6E76657274)
```

Real Time Clock/Calender

by Stefan Keller 533 1612
and Chris Tinney 759 5052

Hello again! Those of you that know which end of a soldering iron to hold might like to try your hand at this Saturday afternoon project. You've all heard of TIME$ (We hope), and you all know how useless it is. (Come on.. how many of you actually type the time and date every time you boot? (LDOS users excepted). A real time clock/calender with battery back up is exactly what the doctor ordered for you lazy typers. This clock is even smart enough to work out that February has 29 days in a leap year.

If you've glanced at the circuit diagram, you will have seen that the circuit is very simple (considering that it is technically quite complex). The clock consists of a MSM5832 clock chip, (the same as the one used in the ETI 'Little Big Board' computer) a 8255 PIO, and 2 small LSTTL decoder chips. The PIO is only half used, so you can run something else from it as well (We are looking for a speech synthesiser to fit the application). In total, 4 units have been sucessfully constructed, 3 using 8255 PIOs, and one using a Z-80 PIO (Others are under construction). To use a Z-80 PIO, you need certain control signals not available on the TANDY 40 pin edge connector. If you own a SCRAP-80, you could (if you desire) use a Z-80 PIO.

Well, lets get straight into a description of the clock chip. It is a CMOS chip and hence a little on the slow side for interfacing directly to a microprocessor running at the speed of our Z-80s, this is why the PIO is needed. It gains us the time that the Z-80 just doesn't want to give. The clock has 13 internal registers and one internal frequency generation function. The registers are for things like SECONDS, MINUTES, HOURS, DAY OF WEEK, DAY IN MONTH, MONTH and YEAR.

The hours function has 2 different modes. (1).. 12 hour mode (with and AM / PM indicator bit). (2).. 24 hour mode (the same as TIME$). The mode desired is selected when the time is first set. It can be changed however, it is just a matter of resetting a bit in the hours*10 register.

The leap year function is not as intelligent as it appears. The 2 sections (ie two 4 bit ports) as an internal function of the PIO. We have used this function, so you are free to use the other 4 bits totally independently for something else. Port C is bi-directional. When you set the clock, it is written to, otherwise the port is a read only port. The addresses, Read, Write and Hold are sent to the clock through port B of the PIO. This port is always a write port. By the way, the Hold function does exactly as its name says... it stops the clock from ticking over.

The Hold function is necessary when you set the time, and optional when you read it. If using a machine language reader, the Hold function is totally unnecessary when reading the time, but still necessary when setting it. A BASIC driver programme needs Hold all the time that the clock is accessed. Don't misunderstand the Hold function, it doesn't stop the clock, just stops it ticking over, so as long as the Hold stays high for less than 1 second (easily done if using machine language) then the time will not be slowed. (We haven't said it yet, but the chip is totally POSITIVE logic.. if that means anything to you).

Now a bit on the software that you'll need to run this thing. Because the chip is not directly connected to the Z-80, you have to muck about a little with sending commands. If for instance, you wish to read from the units section of the seconds counter (register 0), you have to send 16 (0 for the register, 16 for the READ line (And optionally another 64 for the HOLD line)) to port 5. (Port B of the PIO) Then you can read the value through port 6 (Port C of the PIO).

Even though you have to go to all this extra trouble to do anything with the clock, all the hard work has been done by us. We can give anyone that wants it, a copy of a program to set the time up, and another that you can AUTO when you boot that says things like "Good morning! It's 10:34:48 Sunday, 23rd of November 1984". (Well that's what it says at 10:34:48 on the 23rd of November - at other times it will probably be different). This program also resets the value of TIME$ every 6 seconds that the computer is turned on, so all you lazy people don't have to type in a time & date when you boot.

It's written for NEWDOS, and will probably not work under LDOS - any other DOS's are in the Lap of the Gods. If you want some technical data on the clock or PIO chips we can give you some of that too.

After all that verbose crap on what the thing does, here is a little note on what you will have to do to make it. There are only 4 chips, so you can build the lot onto a piece of veroboard, or wire wrap board (as we have done). The size of the lot should be no more than 5cm * 5cm. This is allowing room for the small battery (either 2 mercury or silver cells, or one lithium cell) needed for power down times, and for the voice synthesiser chips that you may wish to add at a later date.

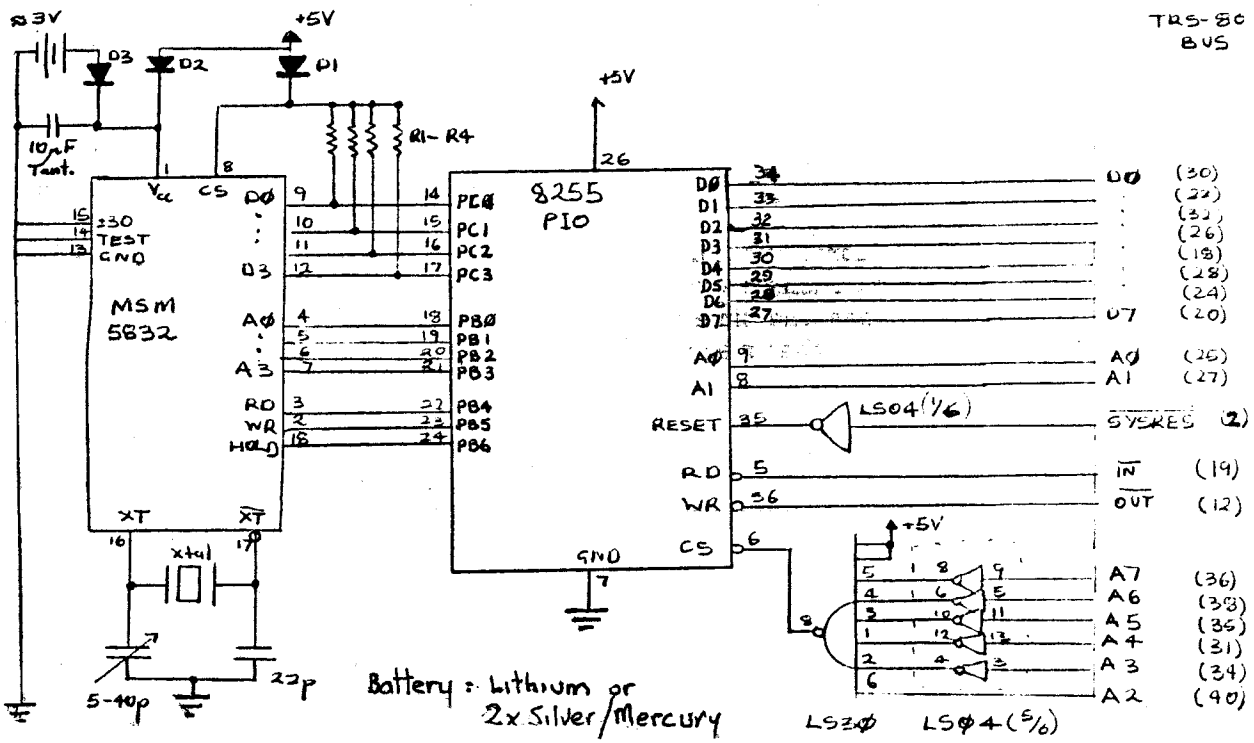
You will have to run around 20 lines back to the expansion bus of the computer (8 data, 8 address, IN, OUT, GROUND, +5 and SYSRES). We have both mounted ours in our (home built) expansion interfaces. We suppose that you could do the same thing with yours once you have built it and verified that it is working. However, you could do as one of our mates at uni has done (not in SYDTRUG)... hang it out the back of his SCRAP-80 on 2 dozen pieces of old wires.

A suggested parts layout is shown. This layout uses one lithium battery. Alternatively use the two silver or mercury cells. For those who want to know where to get parts, we know for sure that Sheridan sells the MSM5832 and the crystal. Other stores probably sell them also.

This is the second lot of articles we have contributed to the newsletter so far, and as you may have realized we are very into hardware mods and assembly language programming. The few mods we have presented so far are only some of ones we have constructed. Others which we have made internal to our CPU's are a DeGlitch (get rid of that video snow), a speedup for Jap CPU's, a Dottizer to turn those adjacent video dots on the screen (which are lines now) into real dots, and a custom Character Generator.

Anyone interested in these mods or having any good ones of their own can see us at the user group meetings (just yell out our names) or give us a call at the above numbers.

If anyone has bothered to build the synthesizer we described several months ago, from the SN76489, then you may be interested to know that I have written a music editor for it. It operates similairly to the FRED program, using EDAS as an editor, and has all sorts of nifty features like CALLS,JUMPS, conditional statements,repeats etc. Any SYDTRUG member who wants a copy merely has to ask! So all of you who want to play around with computer music, and can't afford a pre-built synthesizer - GET HACKING!



Tandy Talks

by Gary Bryce

This column will, I hope, form a regular part of our newsletter, with News, Patches and details of new releases. Just to kick things off this month Alan Harrison of the Tandy Computer Centre at Bankstown gives us some details on the soon to be released Tandy 1000 and 1200 models.

TANDY 1000

The Tandy 1000 comes with 128K RAM, expandable to 640K. Monochrome and Colour graphics adapters are built in (with a RF modulator optional). One double sided double density fourty track disk drive (360k). Interfaces for Printer, Joystick and Light Pen are also supplied. As well as a three voice sound circuit for music generation, plus a speaker. Optional boards include 256K and 512K RAM expansions, RS-232C, Hard Disk Controller and Digi-Mouse/Controller Clock Calendar boards.

The 1000 features three PC compatible internal expansion slots, a 90 key sculptured keyboard with numeric keypad and tactile feedback, Microsoft's MS-DOS 2.11 on an Intel 8088 processor running at 4.77MHz.

TANDY 1200 HD

Meet the Tandy 1200 HD personal computer, it's a "Mirror Image" of the IBM XT - except for the price tag. The 1200 HD comes with 256K RAM, a 360K Floppy disk drive and a 10 Megabyte Hard disk drive. A parallel printer interface is standard and the keyboard features the same layout as the XT, with improved placement of the shift keys.

So for all of you who have been waiting for Tandy to release a true PC compatible, rush down to your local Tandy Computer Centre and bash the door down with your orders.

Review of The Illustrated Multiplan Book

Review by: Peter Wignell.
Author : Russell A. Stultz.
Published by : Wordware Publishing, Inc., Plano, Texas.
Distributed by : Prentice-Hall of Australia Pty. Ltd.
Recommended Retail Price: $27.50

The Illustrated Multiplan Book contains 174 pages of easy to read text and is organised into 25 Modules plus an Appendix. The modules provide command descriptions, applications, and illustrations that sets out to provide an easy to read method of learning Multiplan.

Of the first four modules, Module 1 explains the aim and the organisation of the book. Module 2 describes the Multiplan terms and presents information about how Multiplan can be used to solve everyday problems. Module 3 explains the Multiplan command structure and lists mathematical and logical operators. Module 4 lists a recommended sequence for reading Modules 5 - 25 when learning Multiplan.

Modules 5 - 25 explains each of the main Multiplan commands with each module devoted to one of the commands. The commands described are: - ALPHA, BLANK, COPY, DELETE, EDIT, FORMAT, GOTO, HELP, INSERT, LOCK, LOOKUP, MOVE, NAME, OPTIONS, PRINT, QUIT, SORT, TRANSFER, VALUE, WINDOW and EXTERNAL. Each of these modules starts with a description of the command with two examples, and then gives a list of applications where the command would be used. Finally illustrations are provided showing the use of the command in typical operations. Appendix A provides exercises with questions on each of the modules.

The Illustrated Multiplan Book is easy to read and contains good illustrations of each of the main Multiplan commands but it does not describe in any length any of the mathematical or logical operators. With the many practical examples of the use of Multiplan commands, the book is a worthwhile adjunct to the Multiplan manual but it would not be able to replace the manual. Although the additional command syntax mentioned in the book is CP/M (ie MP B:GPM.MP), the book would be an ideal additional Multiplan reference book for the TRS-80 version of Multiplan.

The Illustrated MULTIPLAN Book

Review by Rowan Evans

When I received this book and began to read it, words like amazing and incredible passed through my mind. Not words of praise for the book, but astonishment that anything so shabby could have been perpetrated on an unsuspecting public. A book so lacking in any redeeming social features that its main use would be for toilet paper!

To begin, the book contains errors of fact in the first module - MULTIPLAN was first produced for the Z80 (CP/M) and the 6502 (Apple Dos 3.3), not the 8086/8088 and MS-DOS (PC-DOS) as is asserted in the section "Hardware and Software Requirements". Then the author goes on to state, "To realise all the benefits of Multiplan, ...at least 64k.... the more memory the better" - this is only true of the 8086/8088 versions, all 8 bit versions will only allow a maximum of 64k to be used.

To read that the book is "an instructionally-designed electronic worksheet textbook" and see that the "recommended learning sequence" jumps from module to module like an epileptic kangaroo is, to me, like telling little Freddy that you pull the pin to make the hand grenade go off, and ignoring to tell him to throw it!

The only chapter that the purported author has concocted appears to be Module Two - "Multiplan Overview", where a heavy crib has been taken of the Multiplan Quick Reference Card to list two pages of control keys and he has written two "sample sessions" that hold ones hand and don't explain a damn thing. Every Module from module 5 onwards, gives an extremely brief discussion of each of the Multiplan menu options and sub-options, a better discussion is available in the Multiplan manual (more closely typed, on smaller pages and containing more and better information - sorry can't have everything) than in this book. There is no discussion on the Multiplan-specific functions and their operation - not merely a serious omission, a fatal one.

In as far as a tutorial text - stick to the Multiplan Manual, in as far as a text on the operations and functions, buy a book like "The Power of Multiplan", it is far better and far cheaper.

In summary, on a scale of 1 to 10, this book rates -99, a definite no-no, hold your nose and pass it by on the other side of the street.

The reviewer has 7 years of experience in the computer industry, the last 3 in Microcomputer Research and was instrumental in having Multiplan adopted as the standard spreadsheet used on microcomputers in the Commonwealth Bank.

Try this Puzzle

by Leonard Yates

On the bench at a recent SYDTRUG meeting were five computers, each of a different type and each using a different brand of diskettes, DOS, printer and modem. During the afternoon, I noted the following things:

1. The LNW80 used Scotch diskettes.
2. DOSPLUS was on a Control Data diskette.
3. An Okidata printer was connected to the Model III.
4. The computer connected to the LPVIII printer used Xidex diskettes.
5. The Model III was immediately to the right of the System 80.
6. The Avtek modem was connected to the computer using NEWDOS/80.
7. The Cicada modem was connected to the Model IV.
8. The C.Itoh printer was connected to the computer in the middle.
9. The first computer on the left was using Verbatim diskettes.
10. The Sendata modem was connected to the computer next to the one using TRSDOS.
11. The Cicada modem was connected to the computer next to the one using LDOS.
12. The Dataphone II modem and the Epson MX80 printer were connected to the same computer.
13. The Tandy Accoustic modem was connected to the computer using Maxell diskettes.
14. The Model I was next to the computer using Verbatim diskettes.

The question is: Which computer was using Multidos and which computer was connected to the Admust printer? (Answer next month.)

Optional NEWDOS/80 Ver 2.0 Zaps

from "Northern Bytes", Vol.5 No.6, where they were reprinted from the CINTUG (Cincinnati TRS-80 User Group) newsletter

These ZAPs to NEWDOS/80 Version 2.0, Models I and III allow the changing of the PDRIVE parameter without writing to disk. The syntax for use is the same as before except "B" is used instead of "A" in the PDRIVE command. The zap allows the PDRIVE table in memory to be changed without changing the disk record.

*** ZAP 091 **** 03/02/84 **** V2M1 ***
SYS16/SYS,02,FB change FE 41 21 20 06 to C3 C0 51 00 00
SYS16/SYS,04,D4 change all zeros to:
FE 41 12 20 03 C3 F1 4F FE 42 C2 F7 4F 3E 41 12 77 C5 E5
01 00 07 21 E2 4D 71 23 10 FC E1 C1 C3 F1 4F

*** ZAP 086 **** 03/02/84 **** V2M3 ***
SYS16/SYS,02,EB change 7E FE 41 12 20 to 7E C3 C8 51 20
SYS16/SYS,04,DF change all zeros to:
FE 41 12 20 03 C3 E0 4F 3D FE 41 20 F8 C5 E5 01 00 07 21
D3 4D 71 23 10 FC E1 C1 18 E3 02

The following ZAP allows you to use the Series I Editor Assembler under NEWDOS/80 Version 2.0 on the Model III. It may also work on the Model I, but it has not been tried out. Remember you must have applied ZAP 025 in order to transfer files from TRSDOS 2.3b over to NEWDOS/80 Version 2.0. The function of this zap is to reload the BREAK enable function of NEWDOS/80. If you are interested on patching other TRSDOS programs, the bytes to look for are 3A C9 32 78 44.

EDTASM/CMD,02,00 change 2A 0D 40 to C3 EA 5B
EDTASM/CMD,05,8D change 4C 69 63 65 6E 73 65 64 20 74 6E 20 to AE 3E C9 32 78 44 2A 0D 40 C3 6B 58

The next patches are for the FORTRAN package and also enable the BREAK key function. They are also for the Model III, and have not been checked for the Model I. Note that FORTRAN/CMD is also called EDIT/CMD on some versions.

FORTRAN/CMD,02,DA change AF 32 1E 53 to C3 A8 7C 00
FORTRAN/CMD,43,54 change 00 00 00 00 00 00 00 00 00 00 00 00 to AF 32 1E 53 3E C9 32 78 44 C3 D2 54
F80/CMD,00,04 change C3 A1 57 00 00 00 00 00 to 3E C9 32 78 44 C3 A1 57
L80/CMD,00,07 change AF 32 15 43 to C3 E6 74 00
L80/CMD,35,72 change 30 00 00 00 00 00 00 00 00 00 00 00 to 3E C9 32 78 44 AF 32 15 43 C3 07 52

Silver Crystal
Prestige in Crystal
Papillon*
Shop G03, Gallery Level
Centrepoint, Sydney 2000
233 4370
* 5% discount to club members on production Membership badge.

(02) 587-3491
R.C.S. RADIO PTY. LTD
Established 1933
IS THE ONLY COMPANY WHICH MANUFACTURES AND SELLS EVERY PRINTED CIRCUIT BOARD AND SCOTCH-CAL FRONT PANEL EVER PUBLISHED IN E.A. & E.T.I./H.E.
651 FOREST RD, BEXLEY, N.S.W. 2207

TRS-80 REPAIR
HYOUNG'S COMPUTER SERVICE
HDS
TEL: (02) 636-8043
101A METELLA ROAD
TOONGABBIE, N.S.W.
2146 AUSTRALIA
HANATRON DATA SYSTEMS

DISCLAIMER

No Patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this publication the Sydney TRS-80 Users Group or it's appointed office bearers assume no responsibility for errors or ommissions. Neither is any liability assumed for damages arising from the use of any information contained herein.