TI*MES

TI99/4A USER'S GROUP (U.K.) CONTACTS.

Chairman: Trevor Stevens. Tel.0623 793077
249 Southwell Rd. East, Rainworth, Nottinghamshire. NG21 OBN
Vice Chairman,Journal Exchange,& Disk Librarian:Stephen Shaw Tel.061 432 6097(8pm+)
(Letters preferred) 10 Alstone Rd.,Stockport, Cheshire. SK4 5AH
General Secretary: Jim Ballinger. Tel.0332 772612
5 Offerton Ave., Derby. DE3 8DU
Membership Secretary, Telecoms,& TI*MES Back Nos.:Peter Walker Tel. 0707873778
24 Bacons Drive, Cuffley, Herts. EN6 4DU
Treasurer: Alan Rutherford. Tel. 0625 524642
13 The Circuit, Wilmslow, Cheshire. SK9 6DA
Publicity Officer: Phil Trotter. Tel. 0642 817356
80 Martonburn Rd., Middlesburgh, Cleveland. TS4 2TH
Programming: Mark Wills.
12 "Rose Hill" Betton St., Belle Vue, Shrewsbury, Shropshire. SY3 7NY
TI*MES Editor & Distribution: Alan Bailey. Tel.081 508 1053
14 Shelley Grove, Loughton, Essex. IG10 1BY
Hardware&Projects: Mike Goddard. Tel.0978 843547
"Sarnia" Cemetary Rd., Rhos, Wrexham, Clwd. LL14 2BY
Librarians:Cassette: Nicky Goddard. Tel.etc. as above.
Modules: Edward Shaw. Tel. 0538 360382(5pm to 8pm)
Crow Holt Farm, Basford, Leek, Staffs. ST13 7DU
Publications: Mike Curtis. Tel. 0209 219051
21 Treliske Rd., Roseland Gdns.,Redruth,Cornwall. TR15 1QE

MAGAZINE CONTENTS

ISSUE NO. 33

SUMMER 1991

EDITORIAL

You will see from the AGM Minutes that we have a new Chairman, Trevor Stevens, who is an active, fully equiped, TIer, centrally placed in the Nottingham area. We welcome him to the Committee, and wish him a long, and happy period in office. It is very encouraging to find that there are still members who will undertake the work involved in keeping our computer alive and kicking!

As you will also see, it is proposed to continue holding a second annual show. Peter Walker has kindly consented to investigate the possibility of arranging it at the conveniently accessible Cuffley venue. As we were lucky enough to have a member from France, with hardware, at the Shrewsbury meeting, perhaps we could hope to attract someone from the active German scene to Cuffley! Does anyone have any ideas?

Two annual shows offer attractive opportunities not to be missed, of having demonstations and tests of software and hardware by their vendors.
Personal experience reinforces my view that it is highly desirable to try before you part with your hard won cash! Any vendor confident of his wares would be happy to demonstrate them, by prior arrangement, at one of these shows. You would also save carriage costs, which in the case of hardware can be considerable, to offset the cost of attending!

DISCLAIMER

Views expressed in this magazine are those of the contributor and not necessarily supported by the Committee. Please don't hesitate to reply if you disagree! We would also like to acknowledge any items from other publications which are not specifically attributed in the text. Let us know and we will put it right.

NEXT COPY DATE

All copy for the next, Autumn, issue of the magazine should reach the Editor by 1st.September. Please make the print as black as possible. and keep it within an area 180mm wide by 262mm high. It makes editing easier if this area starts 15mm from a straight top edge, and 15mm from a straight left hand edge, which are the registration edges for the printers.
You will appreciate that the cost of each page in the hands of the reader is significant, so the more we can include the better. This implies full lines and the avoidance of unnecessary blank areas. Programme listings are an obvious case, where the majority of lines are short, but the occasional long one prevents a compact layout, two columns on the page are preferable. If this causes problems, please limit the lines to 87mm so that cutting and pasting can be used to make up 180mm with a 6mm space between the columns.

TIUG

TI-99/4a USERS GROUP (UK)

MINUTES OF ANNUAL CENERAL MEETING HELD 11th MAY 1991
Held in the MUSIC HALL SHREWSBURY AT 1.00pm.

Stephen Shaw in the Chair

Apologies were received from Trevor Taberner and Derek Hayward

1.Minutes of 1990 AGM (previously published) were approved. There were no matters arising.

2.Stephen Shaw welcomed members to the AGM, pointing out that all officers would be pleased to receive comments regarding the group activities, or to offer advice where required. He was pleased to be able to report that the membership levels had been maintained at a most remarkable level, when one considers that the manufacturer of the TI computer decided to cease production some seven years ago, and that some of the modules were copyrighted in 1978! Yet new programmes and equipment are still being designed for our computer, and anything available when TI retired from the home market is still obtainable through the group.

Much of this equipment is being demonstrated today, and if required could be purchased.

TI*MES is as strong as ever, and the reduction in the number of groups publishing magazines had meant that authors who had been sending text to them were now offering their work to us. As a result TI*MES can now be said to to be at least equal to the best TI mags in the world. He had examples of Micropendium for examination, and would be pleased to arrange delivery for anyone wishing to subscribe.

3.Officer Reports:-

Stephen reported that use of the disc library had declined by about a third, which leaves us a little short of funds. If this continues, it will mean that he must either increase the charges or cease to purchase further programmes as they become available. Searching for and purchasing material is an expensive business, and the cost of a utility disc with programmes by several authors can cost from £50 to £60. The alternative is to exchange discs with other libraries, but this would not encourage the supply of new material into the TI world, and would therefore be counter productive.

Jim Ballinger recalled that during the early years since the Committee had been formed, the formulation and writing up of the constitution and the inevitable snags and amendments that became necessary, together with difficulties each officer met in sorting out their own field etc. made several meetings a year essential. In later years, such matters were sorted out, until last year only one meeting of the Committee was needed to decide on date and location of the AGM, together with the nomination of the officer living in the area involved, to organise this. Barring an unexpected problem, one meeting of the Committee should be all that is necessary this year.

Alan Rutherford reported a most satisfactory year, the total income was £3402.31 including interest, while expenditure has been held to £1375.45 thanks largely to the new terms secured by the editor for the printing of TI*MES, together with prudent expenditure by the rest of the Committee. As a result, the balance of group funds at the end of April 1990 - £1812.07 has increased to £3838.93 in April 1991. It must be pointed out however, that some of this increased balance is due to early re-subscription by many members, with the result that two years subs are shown in the 1990/1 figure in some cases. While it is accepted that increases in the costs can be expected, he felt that there was reason for satisfaction in the present position.

Peter Walker said that he regards the membership this year as most encouraging, when considered against the fall in membership experienced by many groups. During the year 42 members had moved to other machines, but the good news is that during the same period 38 new members joined the group. We are thus able to continue activities, produce our magazine and keep our heads above water, thanks to the efforts of Phillip and Stephen in publicising the group in the UK and USA respectively. We are going through the bulk of the renewal phase at the moment and 38 members who are due to renew at the end of June have not yet done so, and the present level of 166 members may drop then. Things do look fairly good however, and arrangements have been made for the USA members recruited by Stephen to pay their fees in dollar bills, as the cost of changing into Sterling in the USA almost doubles the cost for an American citizen! There are now 18 members of our group who are resident overseas.

He reported that he still had a stock of past TI*MES issues, although his recent offer of 5 issues at £1 each had proved attractive. There had been litle activity in the Telecoms field, though he had done some work for publication, and would gladly give any help he could to anyone with a modem who would like to try exchanging programmes or data via telephone.

Nicky Goddard reported that at the January (Cuffley) meeting he had raised a total of £75, including the raffle since when he has only taken one and a half % of that, which he regarded as disappointing as things were going so well. He had brought plenty of cassettes to the AGM, and would be pleased to serve members present who were interested in a purchase.

Edward Shaw said that the Module Library had quite a good year, with total receipts of £734.20, producing a profit of £124.66. The stock of modules for sale now stands at 77, but the future replacements cannot be certain. The profit balance was transferred today (and so is not shown on the Treasurers account).

Mike Goddard reported that the two most exciting hardware developments, had failed to turn up. The Dutch 80 column card shown at the last AGM had not been available, while the Australian Ramdisc, which he had actually ordered, had been withdrawn because of production problems. He is able to secure most requested items for members, though it takes a little time sometimes. He had secured stocks of TI Joysticks, and cassette leads. Sales of all items were brisk. Stephen mentioned that he had several fliers and catalogues with him - including advertisements for both RAM and GRAM cards by Bud Mills.

Phil Trotter reported that he had, as had been mentioned earlier by Stephen and Peter, been involved in publicity in various ways. He had also been able to draw attention to the Cuffley show, but unfortunately Micromart had not given the publicity he had requested for the present meeting. He would continue to seek ways of putting the TIUG(UK) in the public eye.

Mark Wills reported a reasonably quiet year. He had several letters from members interested in TIbasic aspects, and also from those seeking knowledge of Extended Basic, he was able to help in both fields. He had written both articles and programmes for TI*MES, largely for members with unexpanded systems and would continue to do so, at the same time working on his own software enterprise.

Alan Bailey said that the printers who produced TI*MES for us were quite good, and although their charges were inevitably rising, they were very competitive. Stephen added that, as mentioned in his opening address, the reduced number of publications devoted to the TI had meant that we now had no shortage of material for TI*MES use, and were in the happy position of having many articles awaiting publication space. TIUG(UK) had many services to offer to members, and there could be no doubt that TI*MES was our major attraction.

ELECTION OF OFFICERS

Stephen formally vacated the Chair and asked Jim to conduct the election of the Chairman for the next year.
All officers were elected nem.con.

Office	Nominated	Proposed
Chairman	Trevor Stevens	M.Goddard
Vice-Chairman	Mark Wills	S.Shaw
General Secretary	Jim Ballinger	M.Goddard
Treasurer	Alan Rutherford	N.Goddard
Membership Secretary	Peter Walker	J.Ballinger
Editor/Publisher	Alan Bailey	J.Ballinger
Disk Librarian	Stephen Shaw	M.Wills
Cassette Librarian	Nicky Goddard	P.Trotter
Module Librarian	Edward Shaw	P.Walker

Hardware/Projects	Mike Goddard	M.Wills
Publications Libr.	Mike Curtis	P.Walker
Publicity Officer	Philip Trotter	N.Walker
Programming Officer	Mark Wills	S.Shaw

When the new Chairman was elected, he thanked everyone for their confidence in him, and formally assumed the Chair.

4.Proposed Second Annual Show Peter, who had organised and conducted the show at Cuffley opened the discussion by asking for members to express their views as to the success of the extra show, and a possible repetition. The discussion which followed made it plain that the show was popular, and should be repeated at the same venue - if possible at around the same time of year. Peter agreed to investigate this possibility. The feeling of the meeting was that at Cuffley the ease of access for loading equipment and parking favoured returning there if a booking is available.

The possibility of running a Workshop was muted by Mark, and during the discussion Peter said that he had produced a list showing the concentration of members in each area, remarking that the heavy concentration in the midlands led to that area being selected usually.

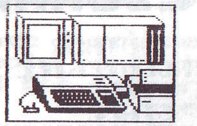
The question of the location of the next AGM was then discussed, Chester, Derby, Birmingham and Stoke were mentioned, but Trevor suggested that this decision could be taken at a later meeting, and this was agreed.

5.Any other Business There being no other business Trevor thanked everyone for attending and declared the meeting closed at 2.00 pm.

J.Ballinger (Gen.Secretary)

Income		Expenditure		
Subscriptions	3014.09	TI*MES publications	898.39	224.6
Back Issues	74.49	Expenses		
		Secretary	39.32	
Disc Library	50.00	Membership	101.57	
		Cassettes	10.37	
Meetings		Modules	70.00	
Door Receipts	72.00			
Raffles	37.37	Hall Hire		
		AGM '90 balance	75.00	
Cassette Library	38.80	Cuffley	50.00	
		AGM '91 deposit	30.80	
Miscellaneous				
Module sale	12.50	Equipment	100.00	
Labels	5.00			
	-------		-------	
			1375.45	
	3304.25			
Bank Interest less charges	98.06	Balance at end April 90	1812.07	
		Balance at end April 91	3838.93	

	3402.31	Surplus of Income	2026.86	



MEMBERSHIP NEWS
and Telecoms Update
by Peter Walker

Since issue 32 we welcome the following new members: Victor Vincent, Robin Russell, Philip Stocks, Jean Louis Cangy, Eddy Carter, William Richardson, James Murta, Michael Barnett, Eric Dyke and Richard Hyland. I hope you all enjoy TIUG.

The Shrewsbury AGM brought together some 22 members, including a new French member Jean Louis Cangy, who showed us some interesting hardware called Maximem and a good deal of software we hadn't seen before. Of interest to me was a telecoms package called ELVIRE, which is a terminal emulation package for the French Minitel videotex system, their equivalent of Prestel. Minitel has been quite a success in France, much more than Prestel in the UK. Minitel terminals have been distributed free as a directory enquiry access device, which has greatly stimulated the system. Independent service providers have also launched a number of dubious services, not unlike the recorded information (0898) services in the UK.

Anyway, the sad thing is that Minitel uses different coding than Prestel, so ELVIRE does not work correctly with Prestel. The graphics and colour codings are not properly interpreted and some text appears in the wrong position, though in general one can make out what is going on. The reason why ELVIRE could be useful to us is that, unlike our own Prestel package (the Dutch Viditel package), ELVIRE offers many features of a proper Terminal Emulation program which many of you have asked me about, namely Print Screen, Save Screen, File Logging, File Upload and much more. There is also a visible cursor, but I can't get the cursor control keys to work, which limits the use of this. I haven't yet analysed everything it can do; I'm told the manual (in French) is very thick. I hope to give more information in a later issue. In the meantime, if anyone is desperate for these features, please get in touch. Below I show some pages printed from ELVIRE to show you what you get.

Peter Walker

TI*MES PUBLICATIONS LIBRARY

LIB.NO.	- TITLE -	- AUTHOR
TPL001	SOFTWARE DEVELOPEMENT	VINCENT/GILL
TPL002	EDITOR ASSEMBLER MANUAL	TI STAFF
TPL003	TI UESER REFERENCE GUIDE	TI STAFF
TPL004	TI USER READ THIS FIRST	TI STAFF
TPL005	USING & PROGRAMMING THE TI	FREDERICK HOLTZ
TPL006	TI FAVORITE PROGS EXPLAINED	DONALD S KREUTNER
TPL007	YOUR FIRST TI99/4A PROGRAM	RODNAY ZAKS
TPL008	BEGINNERS BASIC	TI STAFF
TPL009	TI WRITER MANUAL	TI STAFF
TPL010	TI WRITER TRICKS & TIPS	JOYCE CORKER
TPL011	INTO INTO ASSEMBLY LANGUAGE	RALPH MOLESWORTH
TPL012	[illegible] MANUAL	unknown
TPL013	[illegible] BASIC	HERBERT D PECKHAM
TPL014	[illegible] FOR THE TI	GARRY MARSHALL
TPL015	[illegible]	PETER BROOKS
TPL016	[illegible] WITH THE TI	STEPHEN SHAW
TPL017	[illegible] FOR THE TI	KEVIN TOWNSEND
TPL018	[illegible]	ANDREW NELSON
TPL019	[illegible]	RONALD D ALBRIGHT
TPL020	[illegible] PROGRAMMING GUIDE FOR SPRITES	CRAIG MILLER
TPL021	ENTERTAINMENT GAMES IN BASIC & EXTENDED BASIC (INCLUDES A CASSETTE WITH IT)	KHOA & QUYEN TON
TPL022	THE TEXAS PROGRAM BOOK	VINCE APPS
TPL023	THE BEST OF 99er VOL.1	VARIOUS
TPL024	FUNDAMENTALS OF TI ASSEMBLY LANGUAGE [This is an excellent book for learning about TMS9900 assembly! Mike.]	MORLEY

TI MODULE MANUALS	TI STAFF MEMBERS
TPL025	PHYSICAL FITNESS TRAINING
TPL026	PERSONAL REPORT GENERATOR
TPL027	PERSONAL RECORD KEEPING
TPL028	VIDEO CHESS
TPL029	TOMBSTONE CITY
TPL030	EXTENDED BASIC REFERENCE CARD
TPL031	BASIC REFERENCE CARD

TPL32+ I WOULD LIKE THE LIBRARY TO EXPAND. IF YOU HAVE A MODULE MANUALS NOT NEEDED ANY MORE, PLEASE DONATE OR LOAN THEM TO THE LIBRARY.

Books of the Quarter!

1. TPL024 Learn how to program your ED/ASS or MINI MEM and create your own high speed programs.
2. TPL023 Highlights of Volume 1 of 99er magazine, so much information from those early years!
3. TPL020 Boost your programs with sprites, Craig shows you how to get the most from them!

TI*MES PUBLICATIONS LIBRARY

I have purchased a lot of magazines, books and module manuals out of my own pocket and I am letting these be used for library purposes. The magazines, these are 99er Magazine, an American publication, highly recommended for all aspects of computing with the TI. These will require a deposit of 5.00 each as they are rather hard to come by now, and a hire of 1.00 per month, with a one month maximum time period. I also have a set of Home Computing Weekly, 1.00 deposit per magazine with a 50p hire per month. ALL monies paid for these will go into TIXMES.

NEW BOOKS INCLUDED FOR THE LIBRARYS USE.
9900 FAMILY SYSTEMS DESIGN. - TI Staff
9900 FAMILY DATA BOOK. - TI Staff
PRACTICAL USE FOR THE MICRO IN THE HOME. - David R. Hole
INTRODUCTION TO TI BASIC. - Inman, Zamora, Albrecht
THE ART OF STRUCTURAL PROGRAMMING. - Peter Juliff
SIMULATIONS ON THE MICRO.
BURR-BROWN PCI HANDBOOK. - Burr-Brown
MAKING THE MOST OF YOUR TI99/4A. - Scott Vincent

VAST QUANTITIES OF IC LITERATURE ON TI CHIPS AND SYSTEM CONTROL.

MANUALS FOR MODULES.

Household Budget Management. Personal Record Keeping. Personal Report Generator. Music Maker. Mini Memory. Home Financial Decisions. Terminal Emulator 2. Video Chess. Adventure. Tax/Investment Record Keeping. Statistics. Frogger. Popeye. Q*Bert. Shamus. Defender. Picnic Paranoia. Jungle Hunt. Protector. Donkey Kong. Mind Challengers. Indoor Soccer. The Attack. Return to Pirates Isle. Microsurgeon. Jawbreaker2. Amazing. Fathom. Parsec. Tunnels of Doom. Invaders. Speech Editor. Hopper. Physical Fitness. Car Wars. Blackjack Poker. Yahtzee. Connect Four. Touch Typing Tutor. MXAXSXH.

Many more manuals are expected in the next few months, so updates will be forthcoming!

LIBRARY TERMS.

1. 7.50 DEPOSIT PER BOOK PER MONTH
2. 1.00 HIRE PER BOOK PER MONTH
3. POSTAGE WILL BE BY THE CHEAPEST METHOD UNLESS SPECIFIED OTHERWISE BY THE PERSON REQUESTIN THE BOOK.
4. ALL CHEQUES MADE PAYABLE TO ME AND NOT TIXMES AS I AM RESPONSIBLE TO THE GROUP FOR THIS ENTITY YEARLY AT AUDITING TIME.

MANY THANKS TO ALL WHO HAVE SUPPORTED THE LIBRARY, AND ALL THOSE WHO INTEND TO IN THE FUTURE.
MIKE.

MODULES MODULES MODULES MODULES

APPEAL TO ALL DISK DRIVE OWNERS........

Do you have any modules that you would consider selling or donating to the module library. Reasonable prices paid. For more information please contact me at the address given below.

The latest list of modules available for purchase follows; please note that cheques should be made payable to "E.H.SHAW". Also members are advised to contact me about the modules that they are seeking as the stock is constantly changing.

ADDITION AND SUBTRACTION 1	2.ØØ	PARSEC	3.5Ø
ADDITION AND SUBTRACTION 2	3.ØØ	NUMBER MAGIC	4.ØØ
ADVENTURE and PIRATE TAPE	5.ØØ	HOUSEHOLD BUDGET MAN.	3.ØØ
ALPINER	3.ØØ	HOUSEHOLD MONEY MAN.	1.ØØ
BLACKJACK & POKER	3.5Ø	PROTECTOR	4.5Ø
HUNT THE WUMPUS	3.ØØ	SHAMUS	3.5Ø
BEGINNING GRAMMAR	3.ØØ	DONKEY KONG	4.ØØ
HOME FINANCIAL DECISIONS	2.5Ø	OTHELLO	4.ØØ
CAR WARS	5.ØØ		
CONNECT 4	3.5Ø	THE ATTACK	3.5Ø
MUSIC MAKER	8.ØØ		
EARLY READING	2.5Ø	SPEECH EDITOR	4.ØØ
EARLY LEARNING FUN	2.5Ø	TOUCH TYPING TUTOR	3.5Ø
EXTENDED BASIC AND MANUAL	22.5Ø	* TI MULTIPLAN/DISK/MANUAL	25.ØØ
PERSONAL RECORD KEEPING	3.5Ø	TOMBSTONE CITY	2.5Ø
PERSONAL REPORT GENERATOR	3.5Ø	TERMINAL EMULATOR II	5.ØØ
NUMERATION 1	3.ØØ	VIDEO GAMES 1	3.5Ø
MULTIPLICATION 1	3.ØØ	VIDEO CHESS	4.5Ø
MINER 2049'ER	5.ØØ	YAHTZEE	4.ØØ

* MODULES MARKED WITH AN ASTERISK REQUIRE DISKS OR 32K RAM OR BOTH. ALSO PLEASE NOTE THAT EARLY READING NEEDS A SPEECH SYNTH TO RUN.

PURCHASING MODULES FROM THE LIBRARY

You may return any module purchased within four weeks and be refunded the purchase price less postage which will be charged at the rate of 4Ø pence per module.

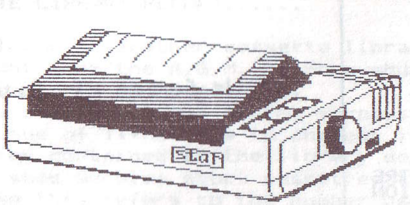
Application to loan/purchase modules.

Name:.............................. Modules required:............
Address:........................... ....
....
....
....

I enclose cheque/PO for £......(as indicated on the list) & post to

PLEASE MAKE CHEQUES PAYABLE TO E.H.SHAW.
Foreign orders can only be accepted if a BANKERS DRAFT is enclosed drawn in STERLING on a LONDON bank. It also helps if a little extra is added on for postage overseas.

MR. E.H. SHAW
CROW HOLT FARM
BASFORD
LEEK
STAFFS. ST13 7DU



PRINTER SURVEY UPDATE

by Peter Walker

Just a few words of update to last issue's printer survey. Some people have had trouble connecting up their printers to the parallel port even when using the wiring connections I printed last time. There could be two sources of difficulty. As I said, it could be that your printer requires a true Centronics interface which calls for a new DSR ROM chip.

However, there may be another simpler solution. Some of the other printer pins may need to be held either high or low for the printer to operate properly. I don't have this problem with my MX80 printer, but the following may be worth trying.

As well as pin 16 being connected to logic ground, the following pins may also need connecting to ground:-

Pin 17 (chassis), though best connected to mains earth connection.
Pins 19 to 30, the return signal for the pins 1 to 12.
Pin 33

In most printers many of these pins are internally connected, so only one may need cross connecting to the TI99. Either pin 11 or pin 16 on the TI99 can be used as ground.

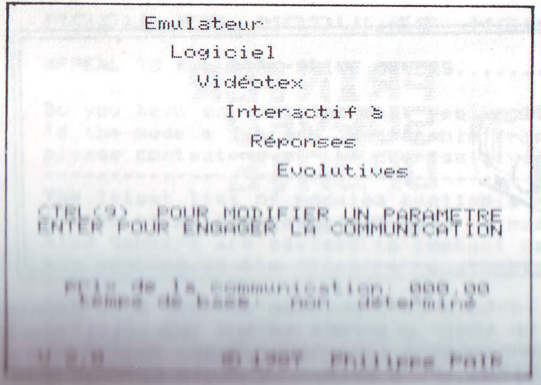
There are two pins which need to be logic "high" for correct operation. Pin 14 controls whether a line feed is automatically added after each line of printing. For use with the TI99, pin 14 needs to be high so that no LF is added. On my printer, pin 14 floats high if left disconnected, but if yours doesn't, then connect to pin 12 on the TI99 to hold it high. On many printers, pin 14 can be held low by DIP switch connection, so ensure this DIP switch is not set.

The same can be said about the Initialise lead on pin 31. The printer is reset if this pin is held low, but usually it floats high when disconnected. However, if necessary it should be held high by connecting to the TI99 pin 12.

Finally, pin 36, Printer Select, needs to be held low, ie connected to Ground. On my printer, a DIP switch holds it low.

I hope this advice helps those still in difficulty with their printer.

Peter Walker



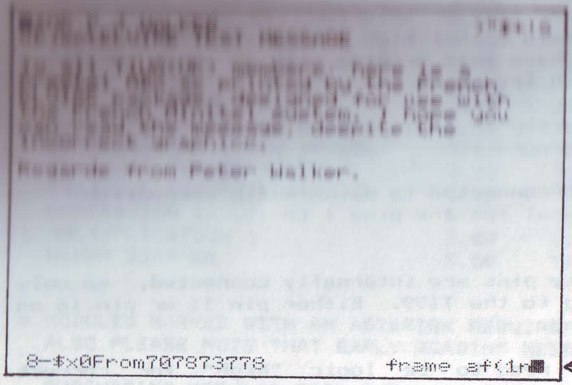
```
Emulateur
  Logiciel
    Vidéotex
      Interactif à
        Réponses
          Evolutives

CTRL(9)  POUR MODIFIER UN PARAMETRE
ENTER POUR ENGAGER LA COMMUNICATION

  Prix de la communication  000.00
   Temps de base   non  déterminé

V 2.0         © 1987  Philippe Pair
```

Title Screen of "ELVIRE"



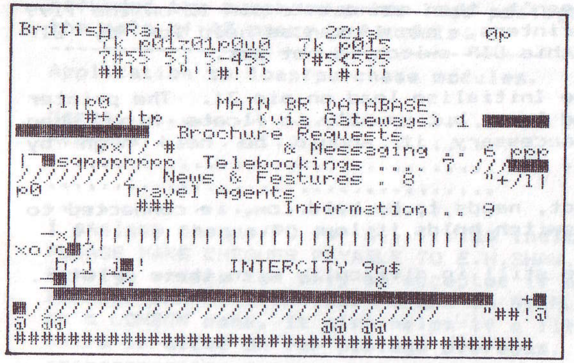
```
[illegible] Test Message                [illegible]
To all TI*USER members, here is a
[illegible] by the French
[illegible] for use with
[illegible] system. I hope you
[illegible] message, despite the
[illegible] graphics.
Regards from Peter Walker.




8-$x0From707873778          frame af(1r▓
```

Reading a Mailbox message

top line appears at bottom



```
British Rail          221a           0p
     7k p01z01p0u0  7k p015
     7#55 5I 5-455  7#5<555
     ##!! !"'!#!!!  ! !#!!!

 ,1|p0       MAIN BR DATABASE
    #+/|tp    (via Gateways) 1  ████
██████  Brochure Requests
    px|/'#        & Messaging .. 6  ppp
|▀█sqppppppp  Telebookings .... 7 //██
/////////  News & Features . 8     "+/1|
p0     Travel Agents
      ###        Information .. 9

  -x||||||||||||||||||||||||||||||
xo/c█?!                d
  h'i i█!      INTERCITY 9n$
 =█|  |  %                &
   ████████████████████████████████  +█
█/////////////////////////////     "##!a
a  aa!                  aa  aa!
############################################
```

British Rail's graphical front p221.

What a mess!

CASSETTE LIBRARY REPORT...... NICKY GODDARD

Hi, welcome to another cassette library report.
I thought that the A.G.M. at Shrewsbury was quite a success and I was pleased to be re-elected.

The postage charge on the surplus stock cassettes went up in the last issue of TI*MES due to changes in the postal rate but prices of cassettes contained in the library does not change.

On some surplus stock cassettes there is a number i.e. G11 on the case this refers to the number used by that particular company NOT the cassette library.

I hope to have the cassette library sorted out soon and I will publish a list in the next issue of TI*MES so keep a look out.

The number of orders has fallen in the past 8 months and although I have some regulars I would like to see more people using the cassette library. I have a fully expanded system with disk drives but I still like to use cassettes.

===

OP PRESS STOP PRESS STOP PRESS STOP PRESS STOP PRESS STOP PRESS STOP

===

SPECIAL OFFER SURPLUS STOCK CASSETTES 50p EACH OR 6 FOR £2.50
POSTAGE 35P FOR THE 1st ONE AND 15p EACH THEREAFTER.

UTILITIES-

TEACH YOURSELF BASIC, BEGINNERS BASIC TUTOR. PERSONAL FINANCIAL AIDS 1, GAMES WRITERS PACKS 1+2(both together), BEGINNERS BASIC TUTOR, TEACH YOURSELF BASIC, CASSFILE, TEACH YOURSELF EXTENDED BASIC, LIST FILE

ADVENTURE-

PYRAMID OF DOOM, SAVAGE ISLAND SERIES, MISSION IMPOSSIBLE, THE GOLDEN VOYAGE, GHOST TOWN.

GAMES-

DADDIES HOT ROD, BATTLESTAR ATTACK. FUNPAC ONE, SOCCERERS CASTLE /LUNAR LANDER, SPACE INVADERS/STARTREK. QUASIMODO. LASER TANK. SENGOKU JIDAI, HUNCHBACK HAVOC. LIONEL AND THE LADDERS. FUNPAC 3, ALONE AT SEA. ADVENTUREMANIA II, SNAKE, PARCO CRAZY CAVER, FORBIDDEN CITY/MASTERBRAIN, HAPPY MATH, PENTATHALON, 99 VADERS, MANIA, ROCK STORM II, TX GRAPHIC PAIRS, HAUNTED HOUSE/WUMPUS. SUPERHOD, OLDIES BUT GOODIES GAMES II, SOCCER SUPREMO, TI TREK, BATTLESHIPS, OLDIES BUT GOODIES GAMES I, VIDEO TITLES, THE CASTLE, ROC.

NB PROGRAMS WITH A STROKE BEFORE THE FIRST LETTER ARE ON THE SAME TAPE AS THE PROGRAM BEFORE IT.

FROM THE PROGRAMMING OFFICER...

Hello again. Yet another rambling (no pun intended Mr. Shaw!) article about programming on our beloved orphan!

Before I get into the nitty gritty can I say a big thankyou to all the members who made the effort to turn up at the Shrewsbury AGM. It was nice to see you all and I hope a lot of you can get down to the workshop in Plas Madoc (details in the last issue) that Mike is organising. I shall be there demoing the software I am selling through Mike (Abbots Software is it's official name) so any one who is interested should pop along.

Well this issue I thought we would look at sound on the 4A in TI BASIC. When the incredibly intelligent (ahem!) programmers at TI developed the TI BASIC and EXTENDED BASIC ROMS for the TI they must have thought something along the lines of "Lets write a CALL SOUND utility that allows the user the use all three sound channels AND a noise channel, but lets do it in such a way that all three channels have to have the same duration so that they can't do proper multi channel music!!!" (yes the sarcasm IS intended!)

Joking apart, the CALL SOUND utility is fine if you want to play multi channel music where all three channels have the same duration (like chords) but if you want to play music where for example a bass note plays continuously and a broken chord plays each of its three notes seperatley over the bass line, then you hit a rather large snag straight away - that is, it is at first sight impossible to play three notes at the same time with different durations - but there is a way around the problem - but it is complicated soloution and therefore will take quite a lot of explaining as to how the soloution works, so bear with me. At the end of this article is a program which you can use in your own TI BASIC or EXTENDED BASIC programs to give you true multi channel music, but you should read the following explanations to find out how it works, and how to use it.

The theory behind the soloution is this: If we put the pitch and duration values in data statements, we can read them into an array, play the note using CALL SOUND for about five seconds (which should cover even the longest of notes) but using a negative duration so that we can interupt the sound at any time during the duration of it. While the note is playing, we will count the duration down in steps of 1 until it reaches zero. When it does get to zero then we will move through the array and get the next note and duration values and play it using the convention described above. It is possible to do this for all three channels of sound and have each of the three channels change notes at different times all through one CALL SOUND statement with a negative duration value which is repeatedly executed.

Let us take an example to clarify matters: Suppose we want to play the note Middle C on channel 1 for 4 "counts" or "beats", an F sharp on channel 2 for 6 beats, and an A on channel 3 for 2 beats. In order to change to next note for each of the three channels at the same time our program must do something along the lines of this:

Play all three notes for a duration of five seconds, (remember we will use a negative duration of about -5000 in our CALL SOUND so that we can change any of the notes without having to wait until it had counted through all 5000 counts) and set up variables for the beat values of each note. Decrement the beat value for the note C and test to see if it is 0. If it is then get the next note from the array and duration value and play them through the call sound. If not zero then check channel 2's beat value to see if it is 0 (the F sharp), is it 0? If yes then play the next note or else check the third notes beat value, to see if it is 0. If it is then get the next note, and play it.

In a nutshell we are taking the note values from the array and playing all three in a CALL SOUND statement then whilst these are playing, we are counting the notes' beat values down and then moving along one in the array and SOUNDing that note whenever the beat value gets to 0. (I said it was confusing.)

Study the program. I have used long self-explanatory variable names so that you can follow what is going on. Type it in and run it, you should get a rather pleasant little polyphonic ditty unless you have made a mistake on the DATA statements!

The DATA statements hold the NOTE then the BEAT values for each note. The first lot of DATA it reads will be played through channel 1, until it comes to a -1, then the next lot of DATA it reads will be played through channel 2 until it comes to another -1, where the next lot will be played through channel 3. A further two -1's after this tells the program that there is no more music DATA to follow.

The program reads the DATA for each channel into its own seperate array at the beggining of the program, (lines 40 - 230), the array MAXN holds the number of elements used in each array so that we can test to make sure we are note trying to play beyond the "end" of our music. If you want to make the program hold more than 100 notes for each channel then change the 100's in lines 20 and 20 to the number of notes you wish to store.

Have fun. Please write to me if you have any problems. I'll be glad to try and sort them out!

Heres the program: Ta Ta for now.

Mark

Mark Wills.

```
10 OPTION BASE 1
20 DIM C1(100,2),C2(100,2)
30 DIM C3(100,2),MAXN(3)
40 CHANNEL=1
50 RESTORE
60 COUNT=1
70 READ NOTE,DUR
75 DUR=DUR+2
80 IF NOTE=-1 THEN 120
90 ON CHANNEL GOSUB 150,180,210
100 COUNT=COUNT+1
110 GOTO 70
120 MAXN(CHANNEL)=COUNT
130 CHANNEL=CHANNEL+1
140 IF CHANNEL=4 THEN 240 ELSE 60
150 C1(COUNT,1)=NOTE
160 C1(COUNT,2)=DUR
170 RETURN
180 C2(COUNT,1)=NOTE
190 C2(COUNT,2)=DUR
200 RETURN
210 C3(COUNT,1)=NOTE
220 C3(COUNT,2)=DUR
230 RETURN
240 MAXN(1)=MAXN(1)-1
250 MAXN(2)=MAXN(2)-1
260 MAXN(3)=MAXN(3)-1
270 REM ** PLAY MUSIC **
280 C1C=1
290 C2C=1
300 C3C=1
310 CALL SOUND(-1000,C1(C1C,1),
    0,C2(C2C,1),5,C3(C3C,1),30)
320 C1(C1C,2)=C1(C1C,2)-1
330 C2(C2C,2)=C2(C2C,2)-1
340 C3(C3C,2)=C3(C3C,2)-1
350 IF C1(C1C,2)=0 THEN 390
360 IF C2(C2C,2)=0 THEN 420
370 IF C3(C3C,2)=0 THEN 450
380 GOTO 310
390 IF C1C=MAXN(1)THEN 360
400 C1C=C1C+1
```

```
410 GOTO 360
420 IF C2C=MAXN(2)THEN 370
430 C2C=C2C+1
440 GOTO 370
450 IF C3C=MAXN(3)THEN 310
460 C3C=C3C+1
470 GOTO 310
480 REM  ** MUSIC DATA **
490 REM  ** CHANNEL 1  **
500 DATA 220,1,233,1,262,2,440,2,
349,2,392,1,349,1,349,1,330,2,330,2
510 DATA 196,1,220,1,233,2,392,2,
330,2,349,1,330,1,294,2,262,2,262,2
520 DATA 220,1,233,1,262,2,349,1,
392,1,440,2,392,1,349,1,
294,2,392,1,440,1,466,2
530 DATA 440,1,392,1,262,2,466,2,
440,2,392,2,349,6,-1,-1
```

```
540 REM  ** CHANNEL 2 **
550 DATA 220,1,233,1,262,2,523,1,
466,1,440,2,440,2,466,1,440,1,
392,1,349,1,392,1
560 DATA 196,1,220,1,233,2,196,1,
175,1,165,2,165,2,175,1,165,1,
147,1,165,1,131,2
570 DATA 220,1,233,1,262,2,523,1,
466,1,440,2,440,2,587,1,523,1,
466,1,440,1,466,2 ,466,2,-1,-1
999 DATA 175,100,-1,-1
```

Notes On Looping:::

No, not the ones in your Garden, the ones of the FOR-NEXT kind! I had a letter from Ken Hughes of Bristol (Hi Ken! Hope I was of help!) who was having a problem with a program from issue 12 (before I joined the group). He told me the computer was bombing out with FOR-NEXT NESTING error, but the line it was objecting to had nothing to do with any loops at all!!

At first me thinks "Aha it must be jumping INTO a loop then hitting a NEXT that it had no FOR for"(?!!) alas this was not so. Much head scratching, various expletives, and deliberately bugged "test" programs later, I arrived at the following theory...

If you take a program like this...

```
10 FOR I=1 TO 16
20 PRINT A;
30 CALL SCREEN(A)
40 REM THE END
```

The computer would report FOR-NEXT NESTING IN 40. Confusing as line 40 does nothing at all!! So what I deduced was this.

When the computer does a "Pre-Scan" (the delay BEFORE your program actually starts where it reserves memory for all your variables) when ever it encounters a FOR command eg FOR PAUSE=1 TO 500 it looks down the rest of the program to make sure there is a NEXT PAUSE somewhere. If not then it refuses to run. But why the strange line number in the error report of the above bugged program? Because during the Pre-Scan the machine encountered a FOR (in line 10) so it searched all the way down the program for the associated NEXT. It didn't find it of course, so it gave the number of the last line in the program, because it got all the way to the end without finding anything.

The moral of this tale? The chances are you are missing a NEXT statement whenever you see this error. Check to see that you havn't mistyped the control variable name in a next statement in your program!

If you ask me, TI should have named this error "FOR WITHOUT NEXT"!!

Thats all on looping. Would ALL members please note that I no longer live at the address on the front cover. I have moved in with my girlfriend. The address on the cover is my parents, so don't worry if you have sent anything there, I WILL get it!

My new address is on the front cover. NO phone! Please don't phone my parents.

A SAD FAREWELL...

A few members of the group will have learned, either through the grapevine or by being a member of EA 99'ers, that Scott and Jo Ann Copeland have regretfully decided to quit the TI-99/4A scene and consequently finish running EA 99'ers. This of course is very sad. I have never met Scott or Jo Ann, and they certainly won't know who I am, but I have learned that they have been very active 99ers over the years, and without question have played a very important part in helping to keep the TI alive all this time. (The TI is now ten years old!)

This of course leaves the EA 99ers as a group with a major re-organisation of personnell on their hands, and I'm sure they will cope with this with no problems at all. I do know however that support for the group is dwindling, and that they may have to pay the expenses for the last issue of this year out of their own pockets, such is their financial situation. I feel this is very alarming for the entire TI "family" across the country - we all know the TI is an orphan, and that if it was not for the user groups none of us would know each other, no new software for our machine would be developed, ditto hardware, it would be impossible to service our machines in the event of a breakdown, and so on. All of which leads to fact that if it was not for the user groups, the TI would be DEAD. There is no two ways about it. I for one would have no interest in writing software for a machine if I thought I was the only one who was going to see it!!

We must support EA 99ers as much as we can. I do not want to see another user group go down the tubes, it would be another nail in the TI's coffin as far as i'm concerned, and I'm sure many others would agree with me. I have written to Mike Curtis to tell him of my sadness that Scott and Jo Ann are leaving, and also to offer him any help that I can. I have joined the group and will be writing articles for them also. I would urge any members that are not members of the group to join, and to send in articles if they can. They publish an A4 magazine of around twenty pages every month, and have a Disk and Cassettee library just like TI*MES does.

I any one would like to know more about the group then write to Mike Curtis, his address is on the front cover.

Support this group. You are letting yourselvs down, and your machine down if you don't.

I will end this sad document by wishing Scott and Jo Ann every luck and success in the future.

Long live the TI. Heres to another ten years!!

Mark

Mark Wills.
Programming Officer

ON THE HARDWARE FRONT....... MIKE GODDARD

Not a lot to report this quarter except that HV99'ers returned my cash and informed me that they have stopped production of the Quest RAM disk because of production problems and once the current batch of disks are sorted out no more will be produced another good product bites the dust!!.

In this issue are some hardware tips culled from other publications while looking for these I came across an article on changing the timing crystal on the main console circuit board the to gain an increase in speed from 12.00 MHz to 14.3181 MHz but on investigation 14.3181 MHz crystals do not seem to be available in this country and I an advised on extremely good authority that such a modification can cause irrepairable damage to the processor and the small increase in speed of approximatly 20% hardly seems worth the risk.

A similar principal applies to the regulators on the power supply boards. Various upgrades for the boards for the console and the PEB have been published in the past to reduce over-heating etc most of which attempt to reduce the over voltage to the boards which in effect is necessary as the type of voltage regulator used requires at least four volts above the required output voltage to operate efficiently. The easiest remedy is simply to parallel another regulator of the same type with the one mounted on the board (with a heatsink if necessary) no additional components are required and Bobs your uncle you have doubled the output of your power supply.

Monitor connections have also become of interest lately when somebody who was trying to sell me a Memotech monitor interface mentioned that it worked extremely well on a mono monitor then I explained that an interface isn't nessary to connect a mono monitor to the TI as it gives an acceptable composite mono signal from pin 2 of the modulator connection of the console. However the Memotech unit does produce a very good composite COLOUR PAL signal and in fact I am now using the same unit to feed the composite input of a TV/monitor with very good results.

Another quick tip for those attempting to change the module socket on the console if you don't have the means to heat all of the pins at once then cut the main body of the connector away from the board by cutting through the pins with a junior hacksaw or similar. Then de-solder the remains of the pins individually which should help stop damaging the circuit board or possibly save your sanity as one chap phoned to tell me the D___N thing WON'T come out not even with a HAMMER !!.

Thats about all from this end of the country if anybody would like to me to cover any specific subject please let me know. Or of course if anyone would like to contribute a Hardware article please do so.

Mike

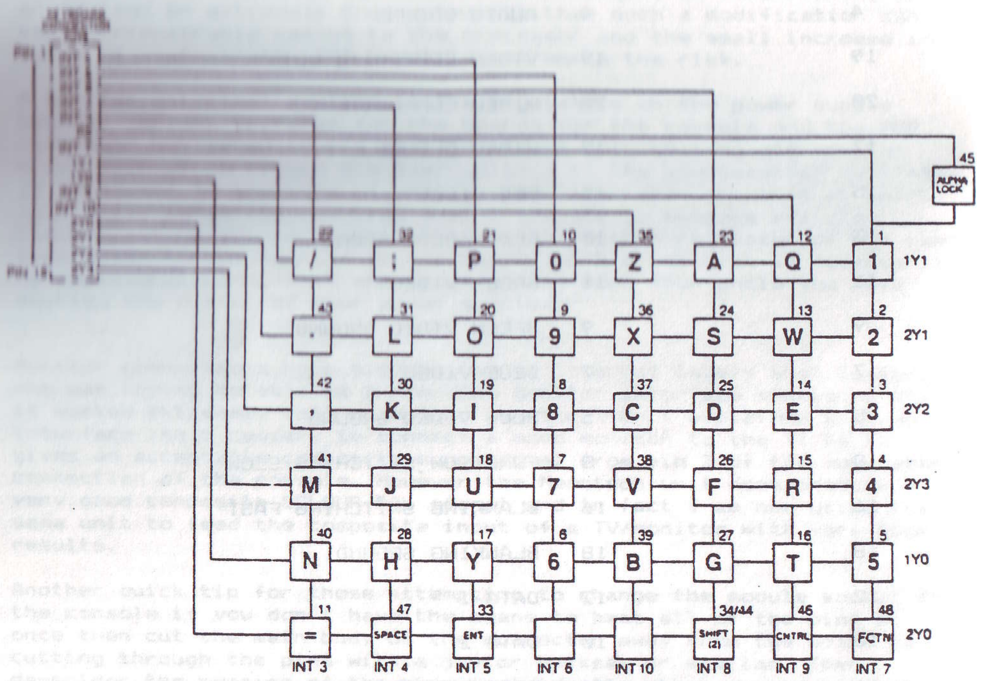
PERITEL (SCART) CONNECTIONS

CONNECTOR 1	CONNECTOR 2	
3	3	AUDIO OUT A
1	1	AUDIO OUT B
6	6	AUDIO IN A
2	2	AUDIO IN B
4	4	AUDIO GROUND
19	19	VIDEO COMPOSITE OUT
20	20	VIDEO COMPOSITE IN
17	17	VIDEO GROUND
15	15	RED VIDEO
13	13	RED VIDEO GROUND
11	11	GREEN VIDEO
9	9	GREEN VIDEO GROUND
7	7	BLUE VIDEO
5	5	BLUE VIDEO GROUND
8	8	FUNCTION SWITCHING SLOW
16	16	BLANKING SWITCHING FAST
18	18	BLANKING GROUND
12	12	DATA 1
10	10	DATA 2
14	14	DATA GROUND
21	21	COMMON GROUND

These connections are reproduced for those who have a TV/monitor with a built in SCART socket and wish to connect a Memotech interface or similar to it in this case the COMPOSITE video signal from the interface should go to pin 20 and earth to pin 17 AUDIO should got to pin 6 and ground to pin 4. This diagram shows how to connect two scart connections together ie TV and VCR etc but for this application just ignore one connector.

Mike

TI 99/4A QWERTY keyboard with standard typewriter stagger format, electrically arranged in X-Y matrix. All keys SPST momentary contact except for ALPHA LOCK, which is an alternate action switch. Electrical connections may be made to connector (AMP 1-640441-5) or directly to PC board.



KEYBOARD SCHEMATIC

The schematic represents the key action in connecting one signal line to another. For example, pushing Q (key #12) connects line 1Y1 (connector pin #8) to line INT9 (pin #10). Pushing ENTER (key #33) connects line 2Y0 (pin 12) to line INT5 (pin 1).

RS232 CARD REPAIR TIP

Patient :- TI RS232 Card. (PE. Box type)

Symptoms :- RS232 operations function as normal BUT PIO will not output correct characters to printer when called on to do so.

Example:-

PIM TEQT, 1Ø14545890=QUEPTYUIMP-AQD

instead of -

PIO TEST. 123456789Ø=QWERTYUIOP/ASD

This particular case would not return a carriage return and therefore would not line feed the printer.

Cure :- Removal of the suspect 74LS245 Bi-directional buffer designated U3 on the RS232 Card and replacement (optionally in a socket) with a brand new specimen. Cost approx $2.5Ø and the usual disclaimers on risking the health of your equipment apply. ie. Whatever you do is at your own risk.

Results :- Success! My PIO port now communicates with my trusty printer in a dialect that I can now interpret.

Albert Anderson - HV99

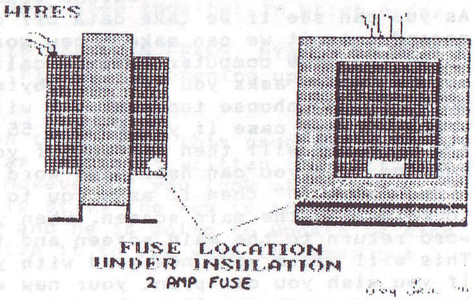
PE BOX POWER SUPPLY

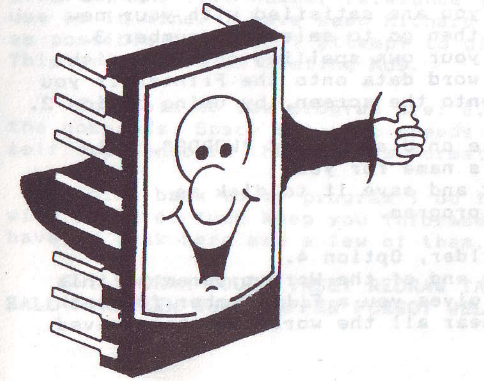
IF YOU HAVE THE MISFORTUNE OF HAVING YOUR POWER SUPPLY QUIT ON YOU, CHECK THE TRANSFORMER VOLTAGE ON THE PRIMARY AND SECONDARY SIDES. IF YOU HAVE THE PRIMARY VOLTAGE AND NO SECONDARY VOLTAGE, THEN CHECK THE FUSE THAT IS LOCATED INSIDE OF THE TRANSFORMER. THE FUSE IS LOCATED ON THE OPPOSITE SIDE FROM THE WIRE CONNECTIONS, AT THE LOWER PART OF THE TRANSFORMER. YOU WILL HAVE TO CUT AWAY THE INSULATION (PLASTIC HOUSING) FROM THE UNIT TO EXPOSE THE FUSE, WHICH IS AN INLINE TYPE THAT IS SOLDERED TO THE WHITE WIRE OF THE PRIMARY SIDE OF THE TRANSFORMER. NEXT CHECK THE POWER SUPPLY BOARD, THE BOTOM LEFT HAND SIDE HAS TWO, ONE AMP DIODES, YOU WILL PROBABLY FIND THAT ONE OR BOTH HAVE SHORTED INTERNALY. CHECK THEM WITH A METER. IF YOU FIND THE VALUES ARE FAULTY, CHANGE THEM.

THE PROCEDURE CAN SAVE YOU FROM A COSTLY REPLACEMENT AND EXTENDED DOWN TIME OF YOU'RE COMPUTER, THE TRANSFORMER AND POWER SUPPLY BOARD COST $127.5Ø FROM TEXAS INSTRUMENTS, NOT TO MENTION THE DELAY FOR SHIPMENT.

RAYMOND LANGEVIN
NORTH EASTERN 99'ers

TRANSFORMER PE BOX





*** THE FANTASTIC WORLD OF SPEECH ***

By T.Stevens   R.Twyning c 1991

I thought it was about time I wrote to you all about some experiments that a friend and I have been doing with speech and its data.

It all started with a look at a little program that was in a 99er magazine a few years back, which I really at the time didnt give a lot of thought about. The program was called verbose. This little program was alright in principle but oh dear what a messy program. I knew I could do better and set about doing a complete rewrite to give it a much better look and feel when it is used.

The program has been split into two parts. Firstly the word Builder part and then the Extractor part. Yes you did read right a word builder. The first program allows you to take two words from the resident vocabulary and build up your own words which can then be saved onto disk for later use.

We will first of all type in the listing **No.1** which you can put onto disk,with the file name "BUILDER". Now having done that you have the first part of the program to play with. As you can see you have various options from 1 - 5. The one we start with is No.1 Join two words.

With "join two words" you must input words that are resident in the speech synthesizer. The first one is the one we are going to reduce the second is going to be added onto it.
To give you an example.. FIRST WORD = READ
SECOND WORD = RIGHT
As you can see if we take data off of the first word and pop the second onto it we can make a new word REWRITE.
To do this the computer automatically checks the length of the first word and then asks you how many bytes you want to take off the first word. If you choose too many, it will tell you and put you back one step. In this case if you choose 55 Bytes it will be about right.
The computer will then ask you if you want to hear the word. By pushing "Y" you can hear your word as many times as you wish.
If you push "N" then it asks you to either redo the byte chopping or to return to the main screen. When you are satisfied with your new word return to the main screen and then go to selection number 3.
This will store your new word with your own spelling onto the disk.
If you wish you can print your new word data onto the Printer if you have one or you can list the data onto the screen, by using option 2.

Fine you have just saved a file on disk called "WORDS". This file was automatically saved in this name for you.
Now if you type in the listing **No.2** and save it to disk as EXTRACTOR we will now discuss this program.

Extractor is run from Word Builder, Option 4.
This second program is the business end of the Word program as this catalogues your word file and also gives you a File Number for each word. If you use option 1 you can hear all the words you have saved.

If you wish to return to Word builder use option 3.
also on this program is option 2 which gives you full instructions on how to use your files in a program of your own choice.
Briefly it allows you with listing **No.3** to merge your word files into any Extended basic program and then use them in progarm.

I am afraid that these programs can't be used to their fullest by the cassette only user. However if he types in the first program he can save the data for putting into his programs by writing out the data on the screen and inputting it again into his programs by hand.

The secret behind all this, is very simple. All you must do is start with the data codes 96,0,X,..... The first of these is to open up the speech synthesizer to except data extraction. The second then sets the internal switches to 0 so that the speech synthesizer knows where it can start. The X is the number of actual bytes that are contained in the data string so that the speech synthesizer knows where to stop. The rest of the data is all allophonic imformation that tells the synthesizer which pitch and slant to use at which procise moment to produce a word. Now I always thought that the voice that comes out is a resident voice and that any other voices that came out were preprogramed into the cartridge they belonged to. Ie the woman on Parsec. Not so,that voice is again all data and will speek direct from the synthesizer with out the cartridge in **EXTENDED BASIC**. If you type in the listing **No.4** you will see what I mean. You can also make it do other things like bark.. yes I did say bark. Try listing **No.5**.

Seeing this program of mine work, my friend Richard Twyning was over the moon at its possible uses. Having more time than me he set about a more complicated Speech Editor right from scratch. This version which follows the same concept as my program, but has the added advantage of being able to put a string of words on to the screen to chop each word about and join them together in which order you liked.
Some of the words he has come up with are TWIN PEAKS, MYARC, RAMBO.
As you can see the world of speech this way is opening up, without the use a TEII type format.

Richard has given me permission to release this program to you. However it isnt fully finished yet as he has'nt written the extractor type program to use it in program. However if you look at the program, the file number reference is still the key to the in program use as in mine. I have seen Richard and he is going to update as soon as possible and I will attempt to give it to you when it comes out.
This program is on listing **No9**.

Controls to use program are: E.S.D.X. to move and Q to action the commands. Space Bar also speeds up down movement. The rest is self explained in the program formatt.

Going back to my program I do intended to update it so I will with a bit of luck keep you informed. Just to show you the words I have on disk here are a few of them.

SWILL DRAM COUGH INSET REDRAW TALL FALL CALL LONELY JAM MANY BALL NINETEEN GORE SUPPER FORGOT WELCOME.

There are hundreds more to make up, so do have some fun. I look forward to any comments regarding this artical or any help in the actual writing of speech codes direct.

Also included into the list of listings are three more speech phrases. No.6 to NO.8. No 7 is a made up composite down loaded from my program and No8 is the same as No 4. but I have edited the zero data out at the end of one and changed the X data count to show you how you can make a word truncate without a program. This however was done with experiment and not knowing quite what I was doing. You can however make up small noises with Richards program save them to file then list the data to printer. I suppose you could take a look at the TE II manual and go through the large list of alaphones. This way you could build up a collection of building blocks to make any word you liked just as TE II does. There is however one area that still puzzels me. That is how do you equate data for other voice types to be produced like in the examples above.
I have looked hard at the data but at this time can find no clue as to how it is done. If there is anyone out there in TI land who knows how this is done I would dearly love to hear from them.

I hope you all have fun with this little lot and find it usfull in your own programs. Be it cassette or disk.

** PLEASE NOTE THAT WHEN A POUND SIGN IS FOUND **
** WRITE THE HASH MARK INSTEAD **

```
100 REM T. STEVENS 1990
110 CALL CLEAR
120 DISPLAY AT(5,5):" +++WORD BUILDER+++"
130 DISPLAY AT(9,1):"ENTER YOUR CHOICE"
140 DISPLAY AT(11,1):" 1 - JOIN TWO WORDS"
150 DISPLAY AT(13,1):" 2 - PRINT SPEECH DATA"
160 DISPLAY AT(15,1):" 3 - STORE NEW WORD ON DISK
170 DISPLAY AT(17,1):" 4 - EXTRACTOR"
180 DISPLAY AT(19,1):" 5 - EXIT"
190 DISPLAY AT(24,1):"ENTER NOW  1"
200 ACCEPT AT(24,12)SIZE(-1)BEEP VALIDATE("12345"):CHOICE
210 IF (CHOICE<1)+(CHOICE>5)=-1 THEN 110
220 ON CHOICE GOSUB 480,770,1280,1550,240
230 GOTO 110
240 CALL CLEAR
250 CALL INIT :: CALL PEEK(2,A,B):: CALL LOAD(-31804,A,B)
260 REM TRUNK 1ST WORD
270 CALL CLEAR
280 DISPLAY AT(22,1):"TRUNCATE HOW MANY BYTES?"
290 ACCEPT AT(22,25)SIZE(3):BYTES
300 MAXBYTES=LEN(B$)-3
310 IF BYTES<MAXBYTES THEN 340
320 DISPLAY AT(22,1):"TOO MANY BYTES SORRY..." :: FOR A=1 TO 300 :: NEXT A
330 GOTO 280
340 IF BYTES>-1 THEN 370
350 DISPLAY AT(22,1):"NO NEGATIVE NUMBERS...." :: FOR A=1 TO 300 :: NEXT A
360 GOTO 280
370 L=MAXBYTES-BYTES
380 C$=SEG$(B$,1,2)&CHR$(L)&SEG$(B$,4,L)
390 RETURN
400 REM
410 CALL CLEAR
420 CALL SAY("",NEWDATA$)
```

```
430 DISPLAY AT(24,1):"SAY AGAIN? Y"
440 ACCEPT AT(24,12)SIZE(-1)BEEP VALIDATE("YyNn"):CHOICE$
450 IF CHOICE$="Y" THEN 420
460 RETURN
470 REM JOIN ROUTINE
480 CALL CLEAR
490 DISPLAY AT(7,1)BEEP:" YOUR FIRST WORD TO JOIN UP :-"
500 ACCEPT AT(8,3)SIZE(15):FIRSTWORD$
510 IF FIRSTWORD$=LASTMADE$ THEN 540
520 CALL SPGET(FIRSTWORD$,B$)
530 GOTO 550
540 B$=LASTDATA$
550 CALL CLEAR
560 DISPLAY AT(7,1)BEEP:" YOUR SECOND WORD TO JOIN UP:-"
570 ACCEPT AT(8,3)SIZE(15):SECONDWORD$
580 IF SECONDWORD$=LASTMADE$ THEN 610
590 CALL SPGET(SECONDWORD$,D$)
600 GOTO 620
610 D$=LASTDATA$
620 CALL CLEAR
630 DISPLAY AT(9,1)BEEP:"ENTER YOUR SPELLING OF THE  NEW WORD AT THE PROMPT"
640 ACCEPT AT(12,6)BEEP:NEWWORD$
650 REM
660 GOSUB 260
670 NEWDATA$=C$&D$
680 GOSUB 400
690 DISPLAY AT(5,1):"    1 - CHANGE SOME MORE":"    2 - BACK TO MAIN MENU"
700 DISPLAY AT(24,1):"NUMBER  1"
710 ACCEPT AT(24,9)SIZE(-1)BEEP VALIDATE("12"):CHOICE
720 IF (CHOICE<1)+(CHOICE>2)=-1 THEN 690
730 IF CHOICE=1 THEN 660
740 LASTMADE$=NEWWORD$
750 LASTDATA$=NEWDATA$
760 RETURN
770 CALL CLEAR
780 DISPLAY AT(5,5):" +++WORD BUILDER+++"
790 DISPLAY AT(11,7):"   1 - PRINT TO SCREEN         2 - PRINT TO PRINTER"
800 DISPLAY AT(24,1):"NUMBER 1"
810 ACCEPT AT(24,8)SIZE(-1)BEEP VALIDATE("12"):INP
820 IF INP=2 THEN 1040
830 CALL CLEAR
840 DISPLAY AT(5,1):"ENTER THE WORD WHOSE DATA   YOU WANT TO SHOW--   "
850 GOSUB 1430
860 CALL CLEAR
870 VALUES$=""
880 PRINT : : :"THE WORD IS ** ";WORD$;" **";
890 PRINT "LENGTH =";L;"BYTES": :
900 PRINT "PRESS ANY KEY FOR NEXT DATA": :
910 FOR I=1 TO L
920 VALUES$=VALUES$&STR$(ASC(SEG$(F$,I,1)))
930 IF I/10<>INT(I/10)THEN 970
940 PRINT "DATA ";VALUES$
950 VALUES$=""
960 GOTO 990
970 VALUES$=VALUES$&","
980 CALL KEY(0,KK,SS):: IF SS=0 THEN 980
990 NEXT I
1000 IF VALUES$="" THEN 2000
1010 VALUES$=SEG$(VALUES$,1,LEN(VALUES$)-1)
1020 PRINT "DATA ";VALUES$
1030 RETURN
1040 OPEN £1:"PIO"
1050 REM
1060 CALL CLEAR
1070 DISPLAY AT(5,1):"ENTER THE WORD WHOSE DATA   YOU WANT TO PRINT--  "
1080 GOSUB 1430
1090 CALL CLEAR
```

```
1100 IF L=0 THEN 1250
1110 VALUE$=""
1120 PRINT £1:"THE WORD IS ** ";WORD$;" **";
1130 PRINT £1:"LENGTH =";L;"BYTES": :
1140 FOR I=1 TO L
1150 VALUES$=VALUES$&STR$(ASC(SEG$(F$,I,1)))
1160 IF I/10<>INT(I/10)THEN 1200
1170 PRINT £1:"DATA ";VALUES$
1180 VALUES$=""
1190 GOTO 1210
1200 VALUES$=VALUES$&","
1210 NEXT I
1220 IF VALUES$="" THEN 1250
1230 VALUES$=SEG$(VALUES$,1,LEN(VALUES$)-1)
1240 PRINT £1:"DATA ";VALUES$
1250 CLOSE £1
1260 RETURN
1270 REM ADD NEW WORD
1280 CALL CLEAR
1290 DISPLAY AT(5,1):"PUT THE DISK WITH ""WORDS""    FILE IN DRIVE ONE." :: FOR
A=1 TO 500 :: NEXT A
1300 DISPLAY AT(9,1)BEEP:"**PRESS ANY KEY TO PROCEED**"
1310 CALL KEY(0,K,S)
1320 IF S=0 THEN 1310 ELSE CALL CLEAR
1330 DISPLAY AT(5,1):"ENTER THE WORD WHOSE DATA   YOU WANT TO SAVE--  "
1340 GOSUB 1430
1350 IF L=0 THEN 1400
1360 OPEN £1:"DSK1.WORDS",INTERNAL,APPEND,VARIABLE 254
1370 PRINT £1:WORD$
1380 PRINT £1:F$
1390 CLOSE £1
1400 RETURN
1410 REM
1420 REM
1430 DISPLAY AT(13,1):">-" :: ACCEPT AT(13,3)BEEP SIZE(15):WORD$
1440 F$=""
1450 IF WORD$="" THEN 1500
1460 IF WORD$=LASTMADE$ THEN 1490
1470 CALL SPGET(WORD$,F$)
1480 GOTO 1500
1490 F$=LASTDATA$
1500 L=LEN(F$)
1510 RETURN
1520 PRINT WORD$
1530 PRINT F$
1540 RETURN
1550 CALL CLEAR :: RUN "DSK1.EXTRACTOR"
2000 PRINT
2001 PRINT "     ** NO MORE DATA **" :: CALL SAY("NO+MORE+DATA")
2003 CALL KEY(0,K,S):: IF S=0 THEN 2003 ELSE 1030
```

Listing 2   File name EXTRACTOR

```
100 CALL CLEAR
110 DIM WORD$(40),F$(40)
120 DISPLAY AT(5,7):" **EXTRACTOR**"
130 DISPLAY AT(15,1)BEEP:"PUT ""WORDS"" FILE IN DISK ONE"
140 DISPLAY AT(24,2):"**PRESS ANY KEY TO START**"
150 CALL KEY(0,K,S):: IF S=0 THEN 150
151 CALL CLEAR :: DISPLAY AT(12,6)BEEP:"WORKING....."
160 OPEN £1:"DSK1.WORDS",INTERNAL,INPUT ,VARIABLE 254
170 FOR I=1 TO 40
180 IF EOF(1)<>0 THEN 240
190 INPUT £1:WORD$(I)
200 INPUT £1:F$(I)
210 NEXT I
220 LAST=I
230 GOTO 250
240 LAST=I-1
250 CLOSE £1
255 GOTO 1000
260 CALL CLEAR
270 DISPLAY AT(5,1):"THERE ARE ";LAST;"WORDS"
280 DISPLAY AT(7,1):"HERE THEY ARE...."
290 FOR A=1 TO 500 :: NEXT A
300 CALL CLEAR
310 FOR AA=1 TO LAST
320 PRINT AA;WORD$(AA);",";
400 NEXT AA
401 PRINT : :
405 DISPLAY AT(5,7):" **EXTRACTOR**"
410 DISPLAY AT(24,1):"WANT TO SAY A WORD Y" :: ACCEPT AT(24,20)SIZE(-1)BEEP VALI
DATE("YyNn"):YN$
420 IF YN$="Y" OR YN$="y" THEN 500 ELSE 1000
500 DISPLAY AT(24,1):"SAY WHICH WORD NUM 01" :: ACCEPT AT(24,20)SIZE(-2)BEEP VAL
IDATE(DIGIT):NM
510 CALL SAY("",F$(NM)):: GOTO 410
1000 CALL CLEAR
1010 DISPLAY AT(5,7):" **EXTRACTOR**"
1020 DISPLAY AT(9,1):"1 - SAY A WORD"
1030 DISPLAY AT(11,1):"2 - MERGE FILE INSTRUCTION"
1040 DISPLAY AT(13,1):"3 - WORD BUILDER"
1050 DISPLAY AT(15,1):"4 - RETURN TO MASTER SCREEN"
1060 DISPLAY AT(24,1):"WHICH NUMBER? 1"
1070 ACCEPT AT(24,15)SIZE(-1)BEEP VALIDATE("1234"):DIG
1080 ON DIG GOTO 260,1200,1100,1090
1090 CALL INIT :: CALL PEEK(2,A,B):: CALL LOAD(-31804,A,B)
1100 CALL CLEAR :: RUN "DSK1.BUILDER"
1200 CALL CLEAR
1210 DISPLAY AT(7,1):"YOU CAN MERGE A ""SAY"" FILE  INTO YOUR OWN PROGRAM"
1220 DISPLAY AT(9,1):"BY LOADING THE FILE ""SAYIT"" IN MERGE FORMAT THEN WITH
EXTRACTOR FILE NUMBERS,     REFER TO THE NEW WORDS THUS:"
1230 DISPLAY AT(13,1):"CALL SAY("""",F$(FILE NUMBER))YOU CAN HOWEVER USE RESIDEN
TWORDS AS WELL BY PUTTING    THEM BETWEEN THE """"."
1240 DISPLAY AT(17,1):"LOAD YOUR MERGE FILE WITH:-                              M
ERGE ""DSK1.SAYIT"""
1250 DISPLAY AT(24,1):"MERGE FILE? Y"
1260 ACCEPT AT(24,13)SIZE(-1)BEEP VALIDATE("YN"):MERG$
```

```
1270 IF MERG$="Y" THEN 1290
1280 IF MERG$="N" THEN 1000
```

LISTING 3   Merge File Name SAYIT

```
1 OPEN £1:"DSK1.WORDS",INTERNAL,INPUT ,VARIABLE 254
2 CALL CLEAR
3 FOR I=1 TO 20
4 IF EOF(1)<>0 THEN 10
5 INPUT £1:WORD$(I)
6 INPUT £1:F$(I)
7 NEXT I
8 LAST=I
9 GOTO 11
10 LAST=I-1
11 CLOSE £1
12 REM USE CALL SAY("",F$(NUM)) WHERE NUM = FILE NUMBER ON EXTRACTOR
```

Listing 4   File name CAUTION;SP

```
1 RESTORE 10 ! [ or starting                    line of                     program
 ]
2 READ A,B,C :: A$=CHR$(A)&CHR$(B)&CHR$(C)
3 FOR I=1 TO C :: READ Z :: A$=A$&CHR$(Z):: NEXT I
4 CALL SAY(,A$)! [ , is                         important ]
5 END
10 DATA 96,0,104
11 DATA 14,8,33,197,3,69
12 DATA 138,102,66,85,109,13
13 DATA 217,236,212,242,178,189
14 DATA 20,187,210,34,34,241
15 DATA 82,236,136,72,27,187
16 DATA 77,49,37,173,61,165
17 DATA 22,117,56,85,213,8
18 DATA 21,128,215,83,18,240
19 DATA 214,121,4,34,16,128
20 DATA 219,131,74,17,165,71
21 DATA 120,155,43,101,80,17
22 DATA 25,45,53,84,94,70
23 DATA 164,134,212,88,249,218
24 DATA 87,170,246,212,170,45
25 DATA 24,171,42,48,142,180
26 DATA 116,66,128,40,211,0
27 DATA 0,0,0,0,0,0
28 DATA 241,1
```

Listing 5 file name DBARK;SP

```
1 RESTORE 10 ! [ or starting                 line of                        program
 ]
2 READ A,B,C :: A$=CHR$(A)&CHR$(B)&CHR$(C)
3 FOR I=1 TO C :: READ Z :: A$=A$&CHR$(Z):: NEXT I
4 CALL KEY(0,K,S):: IF S=0 THEN 4 ELSE CALL SAY(,A$):: GOTO 4
5 END
10 DATA 96,0,100
20 DATA 72,226,178,66,228,225
30 DATA 21,84,141,117,224,156
40 DATA 58,83,188,57,3,58
50 DATA 221,79,235,243,83,88
60 DATA 186,74,240,220,33,119
70 DATA 171,43,36,83,240,211
80 DATA 177,210,83,110,36,188
90 DATA 203,137,210,119,199,80
100 DATA 29,154,64,214,59,166
110 DATA 211,148,35,194,106,202
120 DATA 43,70,145,203,163,43
130 DATA 42,11,72,47,162,149
140 DATA 43,86,179,205,201,133
150 DATA 169,75,183,23,202,50
160 DATA 206,19,84,171,108,108
170 DATA 106,74,153,0,0,0
180 DATA 0,0,239,0
```

Listing 7   File name PRSFIRE

```
1 RESTORE 10 ! [ or starting                 line of                        program
 ]
2 READ A,B,C :: A$=CHR$(A)&CHR$(B)&CHR$(C)
3 FOR I=1 TO C :: READ Z :: A$=A$&CHR$(Z):: NEXT I
4 CALL SAY(,A$)! [ , is                      important ]
5 END
10 DATA 96,0,235
11 DATA 16,128,88,67,155,106
12 DATA 138,103,70,200,217,234
13 DATA 209,76,142,140,19,150
14 DATA 91,107,170,118,217,90
15 DATA 194,36,105,211,133,58
16 DATA 133,28,3,116,89,102
17 DATA 1,11,68,192,3,20
18 DATA 96,64,86,174,67,213
19 DATA 211,35,180,24,45,205
20 DATA 236,136,240,100,124,116
21 DATA 187,34,34,82,206,209
22 DATA 93,79,107,47,217,71
23 DATA 247,205,189,188,232,28
24 DATA 221,39,247,282,146,115
25 DATA 116,159,204,215,82,238
26 DATA 210,125,145,28,79,57
27 DATA 67,247,133,115,180,228
28 DATA 12,221,87,174,150,146
29 DATA 220,244,208,168,147,87
30 DATA 106,211,109,146,108,118
31 DATA 201,85,79,186,183,14
32 DATA 217,26,219,27,2,136
```

```
33 DATA 58,132,3,20,0,64
34 DATA 128,64,225,0,1,248
35 DATA 204,53,0,223,143,38
36 DATA 217,27,181,142,183,60
37 DATA 181,169,101,157,152,200
38 DATA 78,101,184,173,96,20
39 DATA 107,232,84,78,155,30
40 DATA 13,87,205,219,158.122
41 DATA 209,147,134,222,78,74
42 DATA 222,32,134,33,150,99
43 DATA 233,132,1,211,176,53
44 DATA 51,74,245,174,97,178
45 DATA 205,105,53,43,24,11
46 DATA 247,68,157,237,234,24
47 DATA 87,11,123,85,87,165
48 DATA 92,102,242,197,153,181
49 DATA 242,20,211,15,115,197
50 DATA 15
```

listing 8 File Name CAUTION2

```
1 RESTORE 10 ! [ or starting                    line of                         program
 ]
2 READ A,B,C :: A$=CHR$(A)&CHR$(B)&CHR$(C)
3 FOR I=1 TO C :: READ Z :: A$=A$&CHR$(Z):: NEXT I
4 CALL SAY(,A$)! [ , is                          important ]
5 END
10 DATA 96,0,98
11 DATA 14,8,33,197,3,69
12 DATA 138,102,66,85,109.13
13 DATA 217,236,212,242,178,189
14 DATA 20,187,210,34,34,241
15 DATA 82,236,136,72,27,187
16 DATA 77,49,37,173,61,165
17 DATA 22,117,56,85,213,8
18 DATA 21,128,215,83,18,240
19 DATA 214,121,4,34,16,128
20 DATA 219,131,74,17,165.71
21 DATA 120,155,43,101,80,17
22 DATA 25,45,53,84,94,70
23 DATA 164,134,212,86,249,218
24 DATA 87,170,246,212,170,45
25 DATA 24,171,42,48,142,180
26 DATA 116,66,128,40,211,0
28 DATA 241,1
```

Listing 9 File name SP_EDIT

PROGRAM IN TI EXTENDED BASIC

```
1 CALL CLEAR :: CALL INIT :: FOR D=0 TO 14 :: CALL COLOR(D,2,8):: NEXT D :: CALL
 SCREEN(2):: PRINT "":"":"" :: GOSUB 19000 :: ON WARNING NEXT
2 DIM WORD$(9,1),SC(24,32)
3 CALL CHAR(124,RPT$("18",8),126,"000000FFFF",95,"000000FF000000"):: ON ERROR 11
000
4 CALL CHAR(40,"000000030F0C1818",41,"000000C0F0301818",123,"18180C0F03000000",1
25,"181830F0C0000000"):: CALL TITLE
5 CALL OUTPUT(1,1,"~~~~)¦"):: CALL OUTPUT(2,1,"GET ¦¦"):: CALL OUTPUT(3,1,"NEW ¦
¦"):: CALL OUTPUT(4,1,"WORD¦¦"):: CALL OUTPUT(5,1,"~~~~}¦")
6 CALL OUTPUT(6,1,"~~~~)¦"):: CALL OUTPUT(7,1,"SAVE¦¦"):: CALL OUTPUT(8,1,"MAIN¦
¦"):: CALL OUTPUT(9,1,"WORD¦¦"):: CALL OUTPUT(10,1,"~~~~}¦")
7 CALL OUTPUT(11,1,"~~~~)¦"):: CALL OUTPUT(12,1,"SAY ¦¦"):: CALL OUTPUT(13,1,"MA
IN¦¦"):: CALL OUTPUT(14,1,"WORD¦¦")
8 CALL OUTPUT(15,1,"~~~~}¦"):: CALL OUTPUT(16.1,"~~~~)¦"):: CALL OUTPUT(17,1,"TA
KE¦¦")
9 CALL OUTPUT(18,1,"WORD¦¦"):: CALL OUTPUT(19,1,"OFF ¦¦"):: CALL OUTPUT(20.1,"~~
~~}{~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
10 CALL OUTPUT(21,1,"~~~~)(~~~~~~~~~~)(~~~~~~~~~)(~~~~)"):: CALL OUTPUT(22,1,"SEE
¦¦TRUNCATE ¦¦DATA     ¦¦SEE ¦")
11 CALL OUTPUT(23,1,"WRDS¦¦COMPONENT¦¦STATMNTS¦¦ TI ¦"):: CALL OUTPUT(24,1,"~~~~
}{~~~~~~~~~~}{~~~~~~~~~}¦WRDS¦")
12 CALL SPRITE(£1,140,2,10,10,£2,132,7,10,10):: CALL MAGNIFY(3)
13 CALL CHAR(140,"00000001060C081120214080804324180E3146982040C0ACB2C40830C00000
00")
14 CALL OUTPUT(14,31,"(~"):: CALL OUTPUT(15,31,"¦Q"):: CALL OUTPUT(16,31,"¦U")::
 CALL OUTPUT(17,31,"¦I"):: CALL OUTPUT(18,31,"¦T"):: CALL OUTPUT(19,31,"{~")
15 CALL CLS :: GOSUB 15000
16 CALL CHAR(132,"00000000103070E1F1E3F7F7F3C1800000E3860C08000404C38F0C0000000
00FF8181818181FF")
25 CALL SAY("ENTER+YOUR+COMMAND"):: GOSUB 16000 :: IF CC>40 THEN 30
26 ON INT(CR/40)+1 GOTO 1000,2000,3000,4000,5000
30 IF CR>96 AND CR<144 AND CC>240 THEN CALL SAY("FINISHED. GOODBYE"):: CALL CLEA
R :: CALL INIT :: CALL PEEK(2,A,B):: CALL LOAD(-31804,A,B)
31 IF CR>152 AND CC>208 THEN GOTO 8000
32 IF CR>152 AND CC>128 THEN GOTO 7000
33 IF CR>152 THEN GOTO 6000
34 IF WORD$(INT(CR/16),1)<>"" THEN CALL SAY(,WORD$(INT(CR/16),1))
35 GOTO 25
1000 CALL SAY("GET+WORD"):: CALL CLS :: DISPLAY AT(3,6):"PRESS" :: DISPLAY AT(5,
6):"1. FOR TI RESIDENT WORD"
1001 DISPLAY AT(6,6):"2. FOR WORD FROM FILE" :: CALL SAY("ENTER+YOUR+CHOICE")
1002 CALL KEY(0,K,S):: IF S=0 THEN 1002
1003 IF K=49 THEN 1006
1004 IF K=50 THEN 1500
1005 GOTO 1002
1006 CALL CLS :: CALL SAY("ENTER+WORD")
1007 ACCEPT AT(6,6)VALIDATE(UALPHA,DIGIT,"£"):A$ :: CALL SAY(A$):: CALL SPGET(A$
,B$)
1008 GOTO 1600
1500 CALL CLS :: DISPLAY AT(1,6):"GET WORD FROM FILE " :: CALL SAY("ENTER+WORD")
1501 ACCEPT AT(6,6)VALIDATE(UALPHA,DIGIT):A$
1502 OPEN £1:"DSK1.WORDFILE",DISPLAY ,VARIABLE 254
1503 LINPUT £1:I$ :: LINPUT £1:I2$ :: IF A$=SEG$(I$,1,LEN(A$))THEN B$=I2$ :: CAL
L SAY(,B$):: CLOSE £1 :: GOTO 1600
1504 IF EOF(1)THEN CLOSE £1 :: DISPLAY AT(16,6):"WARNING: WORD NOT FOUND":: CAL
L SAY(,WARN$):: CALL SAY(,WARN$):: CALL SAY(,WARN$):: CALL SAY(,WARN$):: GOTO 15
00
1505 GOTO 1503
1600 GOSUB 15000 :: CALL SAY("IN+WHICH+POSITION+DO+YOU+WANT+THE1+WORD")
1601 GOSUB 14000 :: WORD$(LOC,0)=A$ :: WORD$(LOC,1)=B$ :: GOSUB 15000 :: GOTO 25
```

```
2000 CALL SAY("SAVE+WORD"):: GOSUB 17000
2001 CALL CLS :: DISPLAY AT(2,5):" PRESS " :: DISPLAY AT(4,5):" 1. TO VIEW WORDS
2002 DISPLAY AT(5,5):" 2. TO APPEND WORD" :: CALL SAY("ENTER+YOUR+CHOICE")
2003 CALL KEY(0,K,S):: IF S=0 THEN 2003
2004 IF K=49 THEN GOSUB 12000 :: GOTO 2001
2005 IF K=50 THEN 2010
2006 CALL SAY(,WARN$):: GOTO 2003
2010 OPEN £1:"DSK1.WORDFILE",DISPLAY ,APPEND,VARIABLE 254
2500 CALL CLS :: DISPLAY AT(2,6):"HOW SHALL I IDENTIFY" :: DISPLAY AT(4,6):"THIS
 WORD ON DISK?" :: CALL SAY("ENTER+WORD+NAME")
2501 ACCEPT AT(7,6)SIZE(16):N$ :: GOSUB 17000 :: PRINT £1:N$ :: PRINT £1:COMP$ :
: CLOSE £1 :: GOSUB 15000 :: GOTO 25
3000 CALL SAY("SAY+WORD"):: GOSUB 17000 :: CALL SAY(,COMP$):: GOTO 25
4000 CALL SAY("TAKE+WORD+OFF")
4001 CALL SAY("WHICH+WORD"):: GOSUB 14000 :: WORD$(LOC,0),WORD$(LOC,1)="" :: GOS
UB 15000 :: GOTO 25
5000 CALL SAY("SEE+WORDS"):: GOSUB 12000 :: GOSUB 15000 :: GOTO 25
6000 CALL SAY("MAKE+WORD+SHORTER")
6001 CALL SAY("POINT+TO+WORD"):: GOSUB 14000
6002 A$=SEG$(WORD$(LOC,1),4,300):: L=LEN(A$):: CALL CLS :: DISPLAY AT(16,5):"TRU
NCATE HOW MANY BYTES?"
6003 ACCEPT AT(18,5)SIZE(2)VALIDATE(DIGIT):BYTE :: L=L-BYTE :: WORD$(LOC.1)=CHR$
(96)&CHR$(0)&CHR$(L)&SEG$(A$,1,L):: GOSUB 15000 :: GOTO 25
7000 CALL SAY("OUT+PUT DATA"):: GOSUB 17000
7001 CALL CLS :: DISPLAY AT(1,7):"Display to Screen" :: DISPLAY AT(2,7):"or prin
ter " :: DISPLAY AT(4,9):"1. Screen" :: DISPLAY AT(5,9):"2. Printer"
7002 CALL SAY("PRESS+ONE+FOR+SCREEN+OR+TWO+FOR+PRINTER")
7003 CALL KEY(0,K,S):: IF K<49 OR K>50 THEN 7003
7004 IF K=50 THEN OPEN £1:"PIO" ELSE GOSUB 24000
7005 EL=0 :: PRINT £K-49:"DATA ";:: FOR D=1 TO LEN(COMP$)
7006 PRINT £K-49:ASC(SEG$(COMP$,D,1));",";:: EL=EL+1 :: IF EL=8 THEN GOSUB 24020
 :: PRINT £K-49:"":"DATA ";:: EL=0
7007 NEXT D :: PRINT £K-49:"":""
7500 IF K=49 THEN GOSUB 24010 ELSE CLOSE £1
7501 GOSUB 15000 :: GOTO 25
8000 CALL SAY("SEE+£TEXAS INSTRUMENTS£+WORDS"):: RESTORE 20000 :: ROW=1
8001 CALL SAY("PRESS+ANY+KEY+TO+STOP")
8002 READ A$ :: IF A$="*" THEN GOSUB 15000 :: GOTO 25
8003 DISPLAY AT(ROW,5):A$ :: CALL SAY(A$):: ROW=ROW+1 :: IF ROW=20 THEN ROW=1
8004 CALL KEY(0,K,S):: IF S=0 THEN 8002 ELSE GOSUB 15000 :: GOTO 25
11000 ON ERROR 11000 :: CLOSE £1 :: CALL SAY("ERROR"):: GOTO 25
12000 CALL SAY("PRESS+ANY+KEY+TO+STOP"):: ROW=1
12001 OPEN £1:"DSK1.WORDFILE",DISPLAY ,VARIABLE 254
12002 LINPUT £1:I$ :: LINPUT £1:I2$ :: DISPLAY AT(ROW,5)SIZE(24):I$ :: CALL SAY(
,I2$)
12003 IF EOF(1)THEN CLOSE £1 :: CALL KEYPRESS :: RETURN
12004 ROW=ROW+1 :: IF ROW=20 THEN ROW=1
12005 CALL KEY(0,K,S):: IF S=0 THEN 12002 ELSE CLOSE £1 :: RETURN
14000 GOSUB 16000 :: LOC=INT(CR/16):: IF LOC>9 THEN CALL SAY("£THAT IS INCORRECT
£"):: GOTO 14000
14001 RETURN
15000 CALL CLS :: FOR D=0 TO 9 :: DISPLAY AT((D*2)+1,5):WORD$(D,0):: DISPLAY AT(
(D*2)+1,25)SIZE(4):LEN(WORD$(D,1)):: NEXT D
15001 FOR D=2 TO 18 STEP 2 :: DISPLAY AT(D,5):"____________________" :: NEXT
 D :: RETURN
16000 CALL KEY(0,K,S):: IF S=0 THEN CALL LOAD(-31806,64):: CALL MOTION(£1,0,0,£2
,0,0):: CALL LOAD(-31806,16):: GOTO 16000
16001 IF K=ASC("E")THEN CALL LOAD(-31806,64):: CALL MOTION(£1,-10,0,£2,-10,0)::
CALL LOAD(-31806,16):: GOTO 16000
16002 IF K=ASC("X")THEN CALL LOAD(-31806,64):: CALL MOTION(£1,10,0,£2,10,0):: CA
LL LOAD(-31806,16):: GOTO 16000
16003 IF K=ASC("S")THEN CALL LOAD(-31806,64):: CALL MOTION(£1,0,-10,£2,0,-10)::
```

```
CALL LOAD(-31806,16):: GOTO 16000
16004 IF K=ASC("D")THEN CALL LOAD(-31806,64):: CALL MOTION(£1,0,10,£2,0,10):: CA
LL LOAD(-31806,16):: GOTO 16000
16005 IF K=ASC("A")THEN CALL LOAD(-31806,64):: CALL MOTION(£1,0,-20,£2,0,-20)::
CALL LOAD(-31806,16):: GOTO 16000
16006 IF K=ASC("F")THEN CALL LOAD(-31806,64):: CALL MOTION(£1,0,20,£2,0,20):: CA
LL LOAD(-31806,16):: GOTO 16000
16007 IF K=ASC("3")OR K=ASC("4")THEN CALL LOAD(-31806,64):: CALL MOTION(£1,-20,0
,£2,-20,0):: CALL LOAD(-31806,16):: GOTO 16000
16008 IF K=32 THEN CALL LOAD(-31806,64):: CALL MOTION(£1,20,0,£2,20,0):: CALL LO
AD(-31806,16):: GOTO 16000
16009 IF K=ASC("Q")THEN CALL POSITION(£1,CR,CC):: CR=CR+1 :: CC=CC+15 :: RETURN
17000 COMP$=WORD$(0,1)
17001 IF WORD$(1,1)="" THEN 17002 ELSE COMP$=COMP$&WORD$(1,1)
17002 IF WORD$(2,1)="" THEN 17003 ELSE COMP$=COMP$&WORD$(2,1)
17003 IF WORD$(3,1)="" THEN 17004 ELSE COMP$=COMP$&WORD$(3,1)
17004 IF WORD$(4,1)="" THEN 17005 ELSE COMP$=COMP$&WORD$(4,1)
17005 IF WORD$(5,1)="" THEN 17006 ELSE COMP$=COMP$&WORD$(5,1)
17006 IF WORD$(6,1)="" THEN 17007 ELSE COMP$=COMP$&WORD$(6,1)
17007 IF WORD$(7,1)="" THEN 17008 ELSE COMP$=COMP$&WORD$(7,1)
17008 IF WORD$(8,1)="" THEN 17009 ELSE COMP$=COMP$&WORD$(8,1)
17009 IF WORD$(9,1)="" THEN RETURN ELSE COMP$=COMP$&WORD$(9,1):: RETURN
19000 RESTORE 19004 ! [ or starting                  line of
program ]
19001 READ A,B,C :: A$=CHR$(A)&CHR$(B)&CHR$(C)
19002 FOR I=1 TO C :: READ Z :: A$=A$&CHR$(Z):: NEXT I
19003 WARN$=A$ :: RETURN
19004 DATA 96,0,100
19005 DATA 73,227,179,67,229,226
19006 DATA 22,85,142,118,225,157
19007 DATA 59,84,189,58,4,59
19008 DATA 222,80,236,244,84,89
19009 DATA 187,75,241,221,34,120
19010 DATA 172,44,37,84,241,212
19011 DATA 178,211,84,111,37,189
19012 DATA 204,138,211,120,200,81
19013 DATA 30,155,65,215,60,167
19014 DATA 212,149,36,195,107,203
19015 DATA 44,71,146,204,164,44
19016 DATA 43,12,73,48,163,150
19017 DATA 44,87,180,206,202,134
19018 DATA 170,76,184,24,203,51
19019 DATA 207,20,85,172,109,109
19020 DATA 107,75,154,0,0,0
19021 DATA 0,0,240,0
20000 DATA "0","1","2","3","4","5","6","7","8","9"
20001 DATA A,A1,ABOUT,AFTER,AGAIN,ALL,AM,AN,AND,ANSWER,ANY,ARE,AS,ASSUME,AT,B,BA
CK,BASE
20002 DATA BE,BETWEEN,BLACK,BLUE,BOTH,BOTTOM,BUT,BUY,BY,BYE,C,CAN,CASSETTE
20003 DATA CENTER,CHECK,CHOICE,CLEAR,COLOR,COME,COMES,COMMA,COMMAND,COMPLETE
20004 DATA COMPLETED,COMPUTER,CONNECTED,CONSOLE,CORRECT,COURSE,CYAN,D,DATA
20005 DATA DECIDE,DEVICE,DID,DIFFERENT,DISKETTE,DO,DOES,DOING,DONE,DOUBLE,DOWN,D
RAW,DRAWING,E,EACH
20006 DATA EIGHT,EIGHTY,ELEVEN,ELSE,END,ENDS,ENTER,ERROR,EXACTLY,EYE,F,FIFTEEN,F
IFTY,FIGURE,FIND,FINE
20007 DATA FINISH,FINISHED,FIRST,FIT,FIVE,FOR,FORTY,FOUR,FOURTEEN,FOURTH,FROM,FR
ONT,G
20008 DATA GAMES,GET,GETTING,GIVE,GIVES,GO,GOES,GOING,GOOD,"£GOOD WORK£",GOODBYE
,GOT,GRAY,GREEN,GUESS
20009 DATA H,HAD,HAND,"£HANDHELD UNIT£",HAS,HAVE,HEAD,HEAR,HELLO,HELP,HERE,HIGHE
R,HIT,HOME,HOW
```

```
20010 DATA HUNDRED,HURRY,I,"£I WIN£",IF,IN,INCH,INCHES,INSTRUCTION,INSTRUCTIONS,
IS,IT,J,JOYSTICK,JUST
20011 DATA K,KEY,KEYBOARD,KNOW,L,LARGE,LARGER,LARGEST,LAST,LEARN,LEFT,LESS,LET,L
IKE
20012 DATA LIKES,LINE,LOAD,LONG,LOOK,LOOKS,LOWER,M,MADE,MAGENTA,MAKE,ME,MEAN,MEM
ORY,MESSAGE,MESSAGES
20013 DATA MIDDLE,MIGHT,MODULE,MORE,MOST,MOVE,MUST,N,NAME,NEAR,NEED,NEGATIVE,NEX
T,"£NICE TRY£",NINE,NINETY
20014 DATA NO,NOT,NOW,NUMBER,O,OF,OFF,OH,ON,ONE,ONLY,OR,ORDER,OTHER,OUT,OVER,P,P
ART,PARTNER,PARTS
20015 DATA PERIOD,PLAY,PLAYS,PLEASE,POINT,POSITION,POSITIVE,PRESS,PRINT,PRINTER,
SPACE,SPACES,SPELL,SQUARE
20016 DATA START,STEP,STOP,SUM,SUPPOSED,"£SUPPOSED TO£",SURE,T,TAKE,TEEN,TELL,TE
N
20017 DATA "£TEXAS INSTRUMENTS£",THAN,THAT,"£THAT IS INCORRECT£","£THAT IS RIGHT
£",THE,THE1,THEIR,THEN
20018 DATA THERE,THESE,THEY,THING,THINGS,THINK,THIRD,THIRTEEN.THIRTY,THIS,THREE,
THREW,THROUGH,TIME,TO
20019 DATA TOGETHER,TONE,TOO,TOP,TRY,"£TRY AGAIN£",TURN,TWELVE,TWENTY,TWO,TYPE,U
,UHOH,UNDER,UNDERSTAND,UNTIL
20020 DATA UP,UPPER,USE,V,VARY,VERY,W,WAIT,WANT,WANTS,WAY,WE,WEIGH,WEIGHT,WELL,W
ERE,WHAT
20021 DATA "£WHAT WAS THAT£",WHEN,WHERE,WHICH,WHITE,WHO,WHY,WILL,WITH,WON,WORD,W
ORDS,WORK,WORKING,WRITE,X,Y
20022 DATA YELLOW,YES,YET,YOU,"£YOU WIN£",YOUR,Z,ZERO,*
24000 CALL SAY("PLEASE+WAIT"):: FOR R=1 TO 24 :: FOR C=1 TO 32 :: CALL GCHAR(R,C
,SC(R,C)):: NEXT C :: NEXT R :: RETURN
24010 FOR R=1 TO 24 :: FOR C=1 TO 32 :: CALL HCHAR(R,C,SC(R,C)):: NEXT C :: NEXT
 R :: RETURN
24020 IF K=50 THEN RETURN
24021 CALL KEY(0,J,F):: IF F=0 THEN 24021
24022 RETURN
31000 SUB TITLE :: CALL CHAR(132,RPT$("F",16))
31001 PRINT "        (~~~~)"
31002 PRINT "      (~}    {~)"
31003 PRINT "     ()        {)"
31004 PRINT "    ()          {)"
31005 PRINT "    ¦            ¦"
31006 PRINT "   ()            {)"
31007 PRINT "   ¦              ¦"
31008 PRINT "   ¦              ¦"
31009 PRINT "   ¦              ¦"
31010 PRINT "   ¦              ¦"
31011 PRINT "   ()            ()"
31012 PRINT "    ¦            ¦"
31013 PRINT "     ()        ()"
31014 PRINT "      ()      ()"
31015 PRINT "       {~)  (~}"
31016 PRINT "        (~~~~}":"":"    TWYNING ELECTRONICS":""
31017 CALL HCHAR(8,13,132,5):: CALL VCHAR(9,15,132,6):: CALL HCHAR(10,17,132,4):
: CALL HCHAR(13,17,132,3)
31018 CALL HCHAR(16,17,132,4):: CALL VCHAR(10,17,132,7)
31020 PRINT "       SPEECH EDITOR":"" :: FOR D=1 TO 2000
31021 NEXT D :: SUBEND
32000 SUB KEYPRESS :: CALL SAY("PRESS+ANY+KEY")
32001 CALL KEY(0,K,S):: IF S=0 THEN 32001 ELSE SUBEXIT
32002 SUBEND
32765 SUB CLS :: FOR D=1 TO 19 :: DISPLAY AT(D,5):" " :: NEXT D :: CALL VCHAR(1,
31,32,13):: CALL VCHAR(1,32,32,13):: SUBEND
32766 SUB OUTPUT(R,C,P$):: FOR D=1 TO LEN(P$):: CALL HCHAR(R,C+D-1,30):: CALL HC
      HAR(R,C+D-1,ASC(SEG$(P$,D,1))):: NEXT D :: SUBEND
```

Stephen Shaw. 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH
NEW DISKS ADDED since April 1st 1991:

THIS FILE IS MAILED OUT WHEN FULL TO ANYONE WHO HAS PREVIOUSLY SENT AN SAE FOR THE PURPOSE!!!!! KEEP UP TO DATE!

IMPORTANT: DISK LIBRARY CATALOGUE IS NOW MOVING TO FOUR DISKS for a complete set. The FOURTH disk will have the following details taken from the others: GRAPHICS, FORTH, and LOGO. If you only send three disks then you wont get these detailed!

-TIPS. Tips is a new graphics environment and program. The graphics are on a HUGE number of disks- something over 5000 small graphics!!!
>TIPS VN 1.8 by Ron Wolcott. This program allows you to print out all the small graphics in a TIPs graphics file; and, using ten fonts on this disk, and images from the graphics disks, to prepare banners, posters (including calendars), two-fold cards, and labels. Text can be printed in three heights for posters, and the various options between them give you three sizes of image from tiny to huge! Documentation is immense but difficult to follow. You do not need this disk to use the graphics.

>TIPS VN 1.8 SOURCE CODE. No comments but you will enjoy the label names which have been inserted! Added to this disk are tutorials for Vn 1.7 for cards and labels- you can make your way through the CARD tutorial with a little nouse despite the changes in Vn 1.8. Also a novice tutorial for Vn 1.8, by me! And a conversion program from CSGD /GR format to TIPS.

>TIPS PAL. Three programs in c99 (EdAs OP5 load) by T Murphy, which allow you to VIEW all the graphics in a TIPS file, and to convert TIPs images to the more familiar TI Artist INSTANCE format, from which you can convert to almost any other format you can think of! Plus a program by Ernie Pergrem in machine code which will list the names of all the images in a TIPS graphics file, which can typically hold 120 images!

>TIPS IMAGES... well, a huge number of disks(ABOUT 60?). If interested, let me know and I can send a list of image names on disk (FULL disk just for these!), although these are not always useful- Piranha is a tank! Or a VERY brief summary of each disk if you prefer. HOW do you deal with 5000 images in a disk library? Ideas welcome! Printing them all out takes lots and lots of paper! (over 200 sheets I think!).

>GEORGE 8 is now filled with yet another new module to make it to disk, from 1983 the Scott Foresman "per cent" module "SPACE JOURNEY".

>UTIL 27 now full, with a sector editor DISKMASTER,plus the last version of DSKU from John Birdwell- Version 4.2. As John released this version on the basis of NO docs, the library is also retaining vn 4.12 (ask for by name, it fills a disk!). Note that John has since died. DSKU is a good disk manager/sector editor with some nice features. Also, from Ben Yates, DEZIP to unarchive PC files downloaded from BBS in ZIP (archive) format, and PRINT128 to change PC text files from BBS into DV80 or any other display file.

>UTIL 28. All by Ernie Pergrem- programs to manipulate fonts, to move 8x8 Basic defined characters from VDP ram to a disk file for editing or use. An editor for draft or NLQ fonts. And a conversion utility from VDP Ram to TI Artist font files. Apart from the Artist files, the fonts are for use with Epson compatible printers which allow down-load fonts (ESC&, ESC% etc).
>UTIL29. SMALL TALK Vn 3.3 by Lloyd Galenzoski, yes your OWN bulletin board that can run off a single drive system and requires an auto-answer modem! Can handle up to 80 users- ample for the UK. Also from Dave Phillips a program to convert Icon files in the formats: IBM Printmaster, IBM Print Shop and TRS80 Post Master (from BBS in DF128 format). From Ben Yates, TRSSHOW Vn 1.7 which can show pictures in TRS80/Tandy Model III HR and CHR pictures, and convert to TI Artist format. And two utilities from Jesse Slicer, a DSR SCANNER which tells you all you want to know (Myarc ram card is at two addresses!) and a RAM TESTER which requires ram at >6000 and tests all CPU RAM as well as module ram, Myarc 512k ram card and even the ram in a Myarc disk controller.

>The COMPLETE King James BIBLE on disk. Lots of disks. Each chapter in its own file. Each book archived into one file. Fully archived, occupies 13 double sided disks. Some books are over 400 sectors archived! If you dont want the lot in DS archived format, write and ask! Also included are utilities to: Print a verse at a time; to search for words in single verses- eg list every verse with ANGEL in; OR list every verse which has both FISHES and either MANY or FULL. Nifty! Plus for TE2 a program to read the verses to you.

>DRAWMASTER from France, with French docs and windowed choices! Cheaper than TI Artist, but fully compatible with it-loads AND saves TI Artist _P and _C files. Some menu choices appear to be inactive. Press 1,2 or 3 to pull down a menu, and experiment. XB or EA5 load. Unique compact disk format (no not those!) available as an option, saves pictures WITH COLOUR in IV254 files, can be lots less than 50 sectors of TI Artist. No text available as far as I can see, unless you draw it!

NEW DISKS ADDED since MAY 8th 1991:

>GAMES 22 is now full with the addition of Psycho, a math puzzler, and PUZZLE 12, a shape manipulation puzzle, fiendishly hard.

>UTIL 29- managed to squeeze on a tiny utility for use with the Missing Link, which adds turtle graphics, driven from a command line.

>Added to the ADVENTURE disk ADORIA-ATOMSUB is FUN-CITY, a very lengthy Extended Basic adventure for beginners. Real novices. One finger typists. People who can just manage to press ENTER. Claims to be transferred from an PC program.

>BIBLE TRIVIA Vn 3.01 by Steven de Geare, with some nice windowing too! Friendlier than most, you even get points for a wrong answer! but more for a right one. Asks a series of questions and awards a score.

>NOTUNG ADVERT. From Notung Software, their catalogue on disk, plus a little something you cannot put into a printed catalog- an "animated" advert! Disk isued 8 Nov 1990.

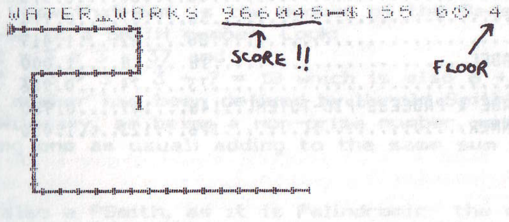
>BEST OF 99ER. FOUR DISKS. From the book of the same name, all the programs ready to use- reference to the book is essential, and if you do not have it, contact the Group's publications librarian.

>SMALL TALK Vn 5.0. The minimum configuration bulletin board software, using modified XModem protocols. TWO DISKS. Requires only one SSSD disk drive with 32k ram and an auto answer modem. Supplied with source code and a small terminal emulator, MINITALK Vn 2.02

>UTIL 30. Not yet full, but contains... RIP OFF a utility to take defined VDP characters and create a TI Artist font file; and ANSI TOOLS Vn 5.2 by T Tesch, a program which allows you to use TI Writer to put together an IBM ANSI file- that is, use the special characters your printer will have available IF it can be switched to "IBM MODE" (Even modern Epson printers have this option). You can create forms and documents using the special characters and print them out, full WYSIWYG as special CHARA1 and C1 files are supplied.

Disks carry a copying fee of just one pound per side, if you send blank disks along! Plus a flat post and packing charge of one pound per order. Cheques payable to S Shaw please. Additional donations of programs and funds to the disk library are welcome- surplus funds are used to seek out and obtain new programs, and further surplus is paid to group central funds.
Donations to Faireware authors may also be made via the libray, please specify author and amount!
A full up to date library listing, on disk in dv80 format, is available in return for four disks and return postage-no copying fee! Overseas members please send several IRCs or US$4 plus disks, or just US$10 or equivalent-currency notes only, no personal cheques from overseas please!

D. Koppenman, 1987. Special to REC Newsletter,



GRAPHICS IMAGES.....

There is a much increased number of programs for graphics on our computer right now, and they do NOT all print out the same!
This article is addressed to answering the questions:
1. How many sizes are available for printouts
2. How much distortion is produced?

The latter point is most valid.... if you drawn a perfect square ON SCREEN how will it look on the printer? If you draw a perfect square by pixel count, how will it appear on a print out?
If you ask most graphics programs to produce a circle, you end up with a rather fat squashed looking oval! If you print that to look like a true circle, then what looks like a square on screen will end up being tall and thin...

First a list of the programs, and some of their various print sizes, marked with an indicator to make listing later less space consuming:
Units of measure are for a sample square. To produce full picture sizes you need to know that a full screen printed as item A below measures 68mm x 119mm.

The first measure given is the height of the square, and if the picture is rotated when printed, this will become-on paper- the width.

The ratio quoted is the height divided by the width, with a low number resulting in a low fat squashed picture, while a high number will produce a tall thin elongated picture.
To produce a good print of a "circle" or pixel-count square, look for a value of between 1.1 and 1.2, while to produce a reasonable copy of what is on screen look for values between 0.97 and 1.00.

	HEIGHT	WIDTH	RATIO
A. TI ARTIST VERSION 2. Print options 1-1-8	36	37	0.973
B. TI ARTIST VERSION 2. Print options 1-2-8	73	74	0.986
C. ARTIST PHOTO 2. SIZE A	18	16	1.125
D. ARTIST PHOTO 2. SIZE B	36	30	1.200
E. ARTIST PHOTO 2. SIZE C	73	61	1.197
F. ARTIST PHOTO 2. SIZE D	110	93	1.183
G. TI ARTIST Vn3. UP. 1/2. 1 pic	70	75	0.933
H. TI ARTIST Vn3. ROTATED. 1/2. 1 PIC	88	59	1.492
I. TI ARTIST Vn3. UP. 1/2. 2 PICS	35	37	0.946
J. TI ARTIST Vn3. UP. 1/2. 3 PICS	35	19	1.842
K. TI ARTIST Vn3. ROTATED. 1/2. 2 PICS	44	29	1.517
L. TI ARTIST Vn3. ROTATED. FULL. 3 PICS	22	59	0.373
M. MAX RLE.....only one size!	36	37	0.973
N. JOYPAINT PAL. NORMAL	36	37	0.973
O. JOYPAINT PAL. DOUBLE	88	62	1.419
P. TIPS 1.8 CARD 3/4 and LABEL	18	18	1.000
Q. TIPS 1.8 CARD-IMAGE	73	74	0.986
R. TIPS 1.8 SIGN-POSTER-IMAGE & PROCESS-PIC-TOTEM	146	149	0.980
S. TIPS 1.8 PROCESS-PIC-BANNER	176	123	1.430

continued........>

Now in order of image height:
for sampled square

18mm	C,P
22	L
35	I,J
36	A,D,M,N
44	K
70	G
73	B,E,Q
88	H,O
110	F
146	R
176	S

And in order of ratios:

0.373	L
0.933	G
0.946	I
0.973	A,M,N
0.980	R
0.986	B,Q
1.000	P
1.125	C
1.183	F
1.197	E
1.200	D
1.419	O
1.430	S
1.492	H
1.517	K
1.842	J

All measurements taken to nearest millimetre. Printer friction fed so also allow a little for slippage.

I have not listed the excellent package SMARTCOPY by Alexander Hulpke as that is a very versatile program allowing fine adjustment to printed ratios, especially for printing larger images where pixel mulipliers can be very finely tuned.

++++++++++++++++++++++++++++++++++++++

RAMBLES

for TI*MES 33.

Welcome to another ramble, first my address, which is the same as that of the disk library:
10 Alstone Road STOCKPORT Cheshire SK4 5AH
Your letters and queries are always welcome, and if a direct response is required, an SAE is a help! For a copy of the disk library please send FOUR disks and return postage.

First item of news comes in a note from Dr Clifford Pickover, who advises me that his magazine Chaos and Graphics is now published commercially as a segment of "COMPUTERS AND GRAPHICS" from Pergammon Press. He also says he is to have a further book published by the end of the year, so that is one I shall be looking for.
=========================

SAFE HEX USE DIGITS

You all know that 666 is the number of the Beast, but did you know His name? 666 is a SMITH number. Look:
666 = 2 x 3 x 3 x 37 and
2 + 3 + 3 + 3 + 7 = 18 which is also 6 + 6 + 6 !!!!
A Smith number has been defined by Harold Smith's brother in law, Prof Albert Wilansky, as being a non-prime number, with prime factor digits (excluding one as usual) adding to the same sum as the digits of that number.

666 is also a PSmith, as it is Palindromic- the same backwards as forwards. It is not a Smith brother however as 665 and 667 are neither of them Smiths.

------> continued --->

Remembering that prime numbers are excluded! there are 81 Smiths between 0 and 2000. Can you derive a program to list them?
Can you derive a 3x3 magic square containing only Smith numbers?-columns and rows and diagonals all add up to the same total.

The highest known Smith number has 3681893 digits... a bit beyond our machine perhaps... it has been proven that there is an infinite number of Smiths and PSmiths but the proof is awaited that there are an infinite number of Smith brothers....
==============================

UPDATE- WATERWORKS.

In the last issue I mentioned this disk based game from Asgard, and later flagged it as bugged.

Here is a fuller review....

The disk comes with a short manual, but the bulk of the rules are on disk to be viewed and/or printed out - instructions fill some 49 screens, and each printes out to the usual 24 lines. Thats a lot of docs to print out.

For the avoidance of any doubts, I did print it all out and read it before I played the game!

For those of you who do not know, I have a seven year old in the house, who likes to see new games, and have them explained to him. This makes a pause key useful, but in any case means that initially a little more time can be taken on a game! Having to print out all those docs did not please him too much...

And both he and I were a trifle (well, a lot actually) unhappy when after a fairly lengthy loading procedure, the game locked out on us- a full lock out with all keys inoperative, while an animated sprite ran havoc over the screen....

The game? To put it as simply as possible, you run a pipeline from the top left of the screen to the bottom left. Your pipeline must be complete and leakfree and must pass through an increasing number of marked points. As the game progresses you get more obligatory locations to serve, and a long list of obstacles and hazards.

The documentation indicates a time limit to complete ALL floors and suggests an average time of 12 minutes per floor- ample for the first few floors.

When I play it the game locks out on the first screen after about two minutes. This is NOT a time-out end of life, it is a fully fledged lock out, requiring the game be reloaded to play again. This condition has been confirmed to me by other players. The problem in my opinion is sheer bad programming.

If you work fast enough you can get on to other screens.

I subsequently found that if I obtained T-junction pipe and laid that, although I could then buy the needed end caps to seal off the pipe end NOT going to the termination point, I was unable to place the end stop. This pertains to the first six screens I got up to.

I also found that by a "legal" use of the hammer to destroy pipe wrongly laid, my score leapt upwards somewhat, and instead of a legal maximum of under a thousand I suddenly had a score of over 900,000. Once more, in my opinion a mark of an appallingly written game.

At this point I gave up.

There are numerous other bugs in this program (chewing gum fails to lose its grip after the documented 60 seconds maximum for example) but as rebooting is such a CHORE I have not explored them very thoroughly.

My supplier, Martin Blyth of Database, and (reported by Martin) the publisher, Chris Bobbitt of Asgard Software, consider this game to be of merchantable quality. In my opinion it is not.

It is possibly the most crappy commercial software I have seen in the last five years.
=========================

I hereby retract my recommendation that you obtain goods from Martin Blyth trading as Database. Should you wish to order goods, you should first obtain from Martin a copy of his "returns policy", as otherwise you may be disappointed. Note that due to considerable fluctuation in exchange rates published prices may alter.
=====================================

The above two items replace a much lengthier article, which looked as though it was going to take over TI*MES! Why should I burden you with my troubles? Let's just leave it at the above, and you may make your own decisions! On to other matters....
=====================================

Increasingly I find members are not quite aware of how the disk library operates... not too surprising as we have many new expanded members since this topic was last covered!

All over the world there are TI owners who program. Some of their programs are well worth sharing, although commercial distribution may not be appropriate.

There is a need to hear of these programs, to obtain them, and to encourage more of them!

Keeping in touch with the TI world is becoming increasingly more difficult, as the numbers of owners and programmers decreases, and keeping in direct contact with known active programmers is essential if we are to benefit from their work. This does cost money.

Encouraging programmers to write, by showing appreciation both by comment and by financial reward, is essential if the TI is to continue to thrive. This can cost a lot of money, but not as much as if we had to rely on full priced commercial products for another computer!

Typically, having heard of a useful utility, the disk librarian (me) will write directly to the author, with a suitable financial inducement- typically five to seven pounds- and in due course the disk will arrive for assesssment. IF it is useful the program enters the disk library. NOT all submissions are added. A typical Utility disk, with programs from several authors, may represent expenditure of fifty pounds or more!!!

These funds do not appear. They come from library charges, and the fewer members using the library, the fewer new programs can be added.

There are user groups who spend a lot of time swopping programs with each other. There are difficulties attached to this- often the programs are corrupted or incomplete (the cause of many complaints!); often they are old and long since updated and improved or debugged. More seriously, swapping programs between user groups is NOT an inducement to authors to write programs- they NEED user contact, and financial contributions are always welcome! -and makes it harder for new programs to enter the chain. I will restrain myself from commenting on TI owners who sit around waiting for goodies to arrive without any payment or activity on their part and complain that the TI is dead 'cos nothing arrives! As with all things you get out what you put in. Put nothing in you get nothing out. The UK is already well known as a land of freebooters, contributing little in the way of programs, and (with some notable and honorable exceptions!) making no donations for faireware. If the TI is to maintain itself, we need to put in!
=====================

Also of course, for YOUR User Group to continue to satisfy YOU we all need oodles of feedback... what pleases you, and more importantly perhaps, what displeases you! What would you like to see in TI*MES!
=============

This issue, in addition to text from other sources, my contribution is rather less text and rather more programs, just for a change, and quite a few of them are not graphical either. Do let me know how you feel about these rather "math" based programs. I have tried to add little notes on programming principles to one or two of the listings.

============

In the last issue I reported on the sad death by cancer of John Birdwell. John wrote the disk utility program known as DSKU. His widow, Kathy, has generously donated his equipment and future freeware donations, to establish a memorial fund, with the task of awarding, at their sole discretion, an annual award to the person or company which has made the greatest contribution to the TI-Geneve world in the preceding year.

You may wish to draw to their attention any person or firm you feel deserves an award.

You may wish to send in a freeware donation, and possibly request the documentaion on the latest version (supplied only without documentation)-

Write to: The John Birdwell Memorial Fund
c/o Chicago TI Users Group, P O Box 578341, CHICAGO, IL, USA, 60657.

The AGM of this group, held in Shrewsbury in May, was rather thinly attended. Next year perhaps we can borrow someones garage? It was howevr very pleasing to meet up with one of our overseas members who had travelled all the way from France. Thank you for coming Jean, we hope you enjoyed the trip.

We have previously broached the idea of having more meetings, spread around so that more members could meet together, without response.

Could it be we need not face the expense of a large commercial venue for the AGM in future? Do you really wish not to meet up with fellow users? Your committee is eager to hear your views. How far would you travel to a meeting- or would you not attend one held on your road?

Data required please!

==============================

REVIEW- JOYPAINT / JOYPAINT PAL

About US$10 from Comprodine

In the last issue I put in an ad for this, printed by itself, but somehow managed not to include a review- could have sworn I wrote one!!! Must have fallen off the disk, apologies.

JOYPAINT came out about the time TI Artist did, but sold for a great deal more -more than twice the price.

You already know of my interest in graphics, and I quickly got a copy, only to find it was entirely unusable, not due to a bug in the program, but due to poor design- it was used by means of icons on the far LEFT of the tv screen, and on MY tv screen, those leftmost columns were invisible!

My old tv having died, I now have a super mono monitor from MCGS, which allows me to SEE those left columns, and coming back to the program I find it quite delightful.

Joypaint uses a larger print size than fits on the screen, and this is handled by windowing- as you move off to the side the whole picture moves! The Joypaint disk uses its own unique (large) disk format, but JoyPaint Pal allows you to load and save TI Artist and Graphx pictures as well as Joypaint pics.

------>continued --->

The usual graphic features are there, and Joypaint Pal even allows you to define your own fill patern, although automatic fill is not supported by Pal, you can still produce a tight pattern manually- instead of painting a solid black you paint the pattern. Joypaint has more patterns than TI Artist anyway.

Joypaint does not have much in the way of fonts or instances, using instead a "clip and paste" idea rather like Graphx, although there is an on-board font just as TI Artist Vn 2 had (dropped from Vn 3 alas).

Joypaint has an OOPS reversal function!

At present prices, provided your tv allows you to see the far left hand side of the picture, Joypaint is an attractive addition for the graphics user.

===========

ARTIST CATALOGUER... Comprodine, US$10 + postage $4.

This is such a simple idea, but obviously not so easy to implement as we have had to wait so long for it. This program will print out your disks of fonts and instances for TI Artist. Thats it.

So why spend $10? Because this program is CLEVER and will cope with different sized instances, and still fit as many as possible on a line. Beleive me, when you have lots of disks of instances, this program is a beaut, even if you do only use it once for each disk! And it is not at all expensive. Another superb graphic offering from Comprodine.

==========

M BINDOFF GAMES DISK 1- Database. Submitted for review- price not quoted.

This disk contains three extended basic games requiring the use of Joystick 1. In the order they appear on the menu they are:

BOUNDER... you must bounce a ball from a platform with a red top to another with a red top. Then there are the drifting deadly spikes to contend with. An unusual implementation which proves quite difficult. It it perhaps a case of seeing how many red platforms you can land on before you run out of lives, because you will run out of lives!

Fortunately you start with many of them, and MAY be talented enough to collect the odd bonus life, although collecting a bonus is so hard it may cause you to lose a life! ExBas programmers will know how hard it is to maintain full joystick control in a complex program, and there are frustrating periods in this program when there really is nothing you can do, for those fatal fractions of a second, to control the ball. I never did get past screen two... There is minimum use of speech which may cause long delays if you do not have a speech synth.

CAVERUN... this is another dodge the sprites program- as indeed are all on this disk- in this one you work your way around a restricted pathway, collecting goodies and avoiding those drifting sprites. I did find screen 2 very hard, but overall this is probably the easiest game of the three. Sad to report I met with a BAD VALUE program bomb at one stage, so cannot report on it being bug-free. Given a choice of two doors to get to the next screen, one door will dump you back at screen one with no extra cash to get.

JEWEL JACK. Referred to in the program as Bomb Jack. Once more an interesting implementation- it has been a long time since any new XB programs have appeared!- in which you collect bonus points and avoid sprites, and again I was unable to get past screen 2!

-----> continued ---->

Overall these three games are well designed, and have neat graphics and play ideas original to the TI world. I do not know how many disk owners apart from myself play XB games? and perhaps one or more may be released on cassette? As I do not know the price I cannot comment on value for money.
The documentation is brief and on disk, with a reader and printer program provided. The review disk suggested you press "enter" to escape reading the docs mid-way, but that bombed the program in due course.
A warm welcome to Matthew and his programs. Beta testing your own games is never easy (you tend not to make the silly mistakes everyone else will!), and Martin does not make a good beta tester, being unable to find or recognise bugs. A third party prepared to search & destroy bugs and add a final polish would probably improve matters no end.
I notice that two of these games are listed in the East Anglia Region magazine as having been donated to their library, so I remain ignorant of the true status of these games.

=====================

TI CASINO by Ken Gilliland from Notung Software.
Games disk(s). US$15 plus postage $4.
Notung Software, 7647 McGroarty Street, Tujunga, Ca, USA, 91042.

Apart from some new scenarios for the Tunnels of Doom module, this seems to be Ken's first games offering. Ken is a commercial artist and has many good graphics orientated programs for the TI user- naturally the graphics in this offering are superb!
As with all Notung software products, Ken will sell to you in your preferred disk format at no extra cost- if you have the minimum of SSSD then this product comes on two disks, else one DSSD disk.
TI Casino requires joysticks, 32k ram and Extended Basic; a printer is optional.
Betting games are by no means unusual- some of the very first commercial offerings for the TI were betting games, whether we look at TI's ancient Blackjack and Poker module, or the very early cassette version of Blackjack from PRP Computergraphics.
What Ken has done here is to assemble eight betting games into one casino suite, so that you can move from game to game, taking your money with you, very largely under joystick control only.
The graphics are of a very high standard, card shuffles are rapid, and it is profoundly easy to lose money. If you have any winnings, you can draw your cash by means of a printed cheque! Graphics? Well, the cashier has a habit of winking at you...
The games are Blackjack (with doubling down, splits, and insurance); Roulette (with 0 and 00 on the wheel); Craps; Baccarat, Acey Deucey; Keno; Draw Poker; and a rather simplistic one armed bandit (no holds or nudges).
If you enjoy gambling games then you will enjoy this offering. If you enjoy well written programs, you will enjoy this!
Documentation? Outstanding. FORTY EIGHT PAGES. And you don't have to print it from disk- it's ready printed for you. Not that you need the docs, but if you are not a casino habitue you may find the details helpful!

============

CERTIFICATE 99 COMPANION PLUS. Notung Software. US$7 plus $4 post.
Requires: Certificate 99 ($10 from Comprodine).

Ken has applied himself to increasing the graphics for Certificate 99, and this time has added ELEVEN new fonts, thirty new border graphics, 60 new graphics!; six new signatures (including Ken and Thug), and seven pre-saved certificates.

If you are into A4 certificates, this is a helpful addition.

==================

STAR TREK-THE NEXT GENERATION CALENDAR:
Notung Software. $10 + $4 p and p. DISK. Specify format!

This collection of MacFlix graphics which prints out to A4 size calendar sheets with photo-scans of tv characters from the show, has been updated and orders sent in now will run from June-June (eg mid year to mid year). Unfortunately these things only seem to come out a week before the year starts, so I am always going to be too late telling you about them! However, so you miss a month or two! You still have ten months to enjoy...
and the price is reasonable!
This offering comes with its own print routines- no other program required.

===========================

In Issue #30, page 53, a puzzle was set regarding clocks, and in the 1st issue a program was listed showing the puzzle was unsolvable. Walter Allum was written as follows, regarding the further problem of just how close we can get to a solution...

The times when the hands are nearest to equal angles can be determined by finding the minima (lowest value) of:

```
  VAR = (I1-4)^2 + (I2-4)^2 + (I3-4)^2
```

where I1,I2,I3 are the inter hand intervals measured on the hour scale.
VAR is a rapidly oscillating function ranging from 0 (equal angles = first problem solved!), to 96 (all hands at 12).

Two methods of locating the minima were used. One was computer search, using a program that "homed in" on the lowest minima with progressively finer search grids over shorter periods. The other was based on calculus procedures aided by computer.
The nearest approach to equal angles comes in the neighborhood of 2.9096 hours, and the complementary time of 9.0904 hours. At this point VAR takes a value of 0.0000472, and the angles between the hands in degrees are 120.086, 120.082 and 119.832.

If the clock differs from normal, with the second hand going round more than 360 degrees per minute - to be specific, taking 59.6666667 seconds to go all the way round, while the minute hand still moves ove minute per minute, then the perfect solution occurs every time the hour and minute hands form a 120 degree angle. (Refer to last issue for times!).

Walter is happy to take this further should there be a demand for it!

================

GUIDE TO TI'S 32K MEMORY EXPANSION
by Peter Hutchison

Just what is a 32k memory and how does it work and what is it used for? I hope this article will solve these questions.
If you plug in your expansion RAM, on the right hand port in my case, and switch on, plug in Extended Basic and type the command SIZE:

```
 13928 BYTES OF STACK FREE
 24511 BYTES OF PROGRAM SPACE FREE
```

[differing values will be given for disk users depending on state of CALL FILES, and may also differ with some versions of ExBas. sjs].

---->CONTINUED---->

The 13k part is the RAM that comes in the console [VDP RAM] and the 24k part is the Expansion RAM.
Where has the other 8k gone to?
The 8k part is for machine code routines and is not used for Extended Basic.
[Extended Basic can LINK to machine code routines in this area. Pure machine code programs can fill all 32k if required].

Another question is - why am I limited to 12k programs if I use cassette for storage? The answer to that is the way the TI saves code. If you have one, look at the Editor Assembler manual, page 297- SAVE. It says "the SAVE operation writes a file from VDP ram to a peripheral".
How much VDP ram do we have? 13k. The TI in fact copies the PROGRAM from Expansion RAM to VDP RAM and THEN to the peripherals- this explains the short delay when you type SAVE CS1.
[Longer programs are allowed on disk as the console switches to an alternate format which saves programs as shorter records instead of dumping them all at once. This format is 'Internal Variable 254' but you needn't worry about that].

The 32k ram area does not have continuous addresses (Editor Assembler manual page 400):

```
>0000... console ROM ( 2 x 4k rom chips)
>2000... low memory expansion (8k)
>4000... peripheral roms for Device Service Routines
>6000... reserved for modules (8k)-ROM or RAM.
>8000... memory mapped devices, VDP, GROM, SOUND, SPEECH
>A000... High memory expansion (24k)
```

If you use Extended Basic, High Mem will contain your basic program and numeric data, console ram will contain strings, and data for the screen display, sounds, sprites, pattern definitions.
 Low memory on the other hand will be used (if at all) by CALL LOAD and CALL LINK commands, for loading and linking to machine code routines.
Expansion memory can be used by other modules, such as Editor Assembler (32k to store machine code programs), Mini Memory (as Editor Assembler, OR as two ram disks), TI LOGO (32k required), etc .
 With the disk system, 32k is "essential" as the disk operation takes up some memory from VDP RAM.
 32k ram cannot be used by TI Basic or modules not designed for its use.
 [Ram disks may incorporate rather than replace the 32k standard expansion, for example the Myarc 512k card uses 32k for normal purposes, and the remainder for ram disk or printer buffer uses.]

Peter Hutchison
6 Moorlands View, Free School Lane, HALIFAX, HX1 2XQ
=====================

Bill Sponchia presents...
HINTS, TIPS & ANSWERS (HTA)
BASIC & EXTENDED BASIC

1. The following program will load, run and then remove itself from memory and will put you in XB without going to to disk drive #1 looking for the program "LOAD". (You could place it on disk as a dummy LOAD file).

```
100 CALL INIT::CALL LOAD(-31952,255,0,255,0)
110 END
```

2. The following program should be saved in the MERGE format as a handy utility:

```
1  CALL CLEAR
2  OPEN #1:"DSKn.",INPUT,RELATIVE,INTERNAL   !!n = disk drive #
3  INPUT #1:A$,J,J,K
4  DISPLAY "SIZE=";K;" USED= ";J-K
5  FOR LOOP= 1 TO 127
6  INPUT #1:A$,A,J,K
7  IF LEN(A$)=0 THEN 10
8  DISPLAY A$;TAB(12);J
9  NEXT LOOP
10 CLOSE #1
11 INPUT "NAME OF PGM TO DELETE ELSE TYPE GO ":DEL$
12 IF DEL$="GO" THEN 15
13 DELETE "DSKn."&DEL$                         !!n = disk drive #
14 GOTO 11
15 STOP
```

When programming this program can be used to catalog a diskette without getting out and loading in a disk manager. It is also written to allow for the deleting of files from the diskette. To use just merge in with the program you are presently working on (of course it is assumed you started it a line 100 and did not use lines 1 to 15).

3. To turn off the QUIT key (FCTN =): CALL LOAD(-31806,16)
Note: Now you must use "BYE" to quit BASIC and get back to the title screen.

4. In an IF-THEN-ELSE statement if you refer to a variable without any other type of relationship then this means "does not equal zero"
eg - IF X THEN 140 ELSE 100 - means If X does not equal 0 then go to 140 but if it equals 0 then go to 100.

5. When using XB prescan it will enable you to run programs that the computer will normally reject. Two things that I have been able to do are - (a) mixing FOR-TO statements and IF-THEN-ELSE statements under one line number; and (b) input two NEXT's but have only one FOR. Here is the program showing these:

```
100 J=0
110 !@P-
120 FOR J=1 TO 20 :: IF J/2=INT(J/2) THEN PRINT J ELSE NEXT J
130 IF J>19 THEN 150
140 NEXT J :: !@P+
150 END
```

6. To automatically return to the Master Title Screen (or Menu for Vermenu users) instead of "END" insert the following:

```
CALL INIT::CALL PEEK(2,A,B)::CALL LOAD(-31804,A,B)
 or
CALL INIT::CALL LOAD(31804,0,36)
```

7. To restrict a CALL KEY statement to taking only one input at a time, no matter how long a particular key is held down -

```
100 CALL KEY(0,K,S)::IF S<1 THEN 100
```

By restricting the status to +1 this will overcome the problem of the sometimes repeating key.

8. To get TRUE random numbers install this line into your program:

```
CALL PEEK(-31880,A,B)::CALL INIT::CALL LOAD(-31808,A,B)
```

9. To erase the program from memory but not erase the screen (and not disturb any assembly routines in lower memory:

```
CALL INIT::CALL LOAD(-31952,255,231,255,231)
```

10. The manual tells you that there are 16 different character sets that you can redefine and change colors on. Actually there are 17 - Set #0 is never mentioned.

11. When LISTing a program and you see a line reference to "32767" this means (unless you actually used that line #) that you resequenced the program while you had a GOTO (or GOSUB, etc) to a non-existing line.

12. To LIST a portion of a program to the printer then enter the following command: LIST "filename":line number range
eg: LIST "PIO":130-240 - will LIST to PIO lines 130 to 240 (inclusive)

13. deleted. sjs.
14. deleted. repeat of no 3 above! sjs

15. Did you know that you could identify your GOSUB routines within the program without using the "!" or REM statement. You are allowed to put one word (string) after the GOSUB line number. Here's an example program:

```
10 CALL CLEAR::PRINT "HERE I GO.."
20 GOSUB 50 DELAY_ROUTINE ::PRINT "I'M BACK!"
30 END
50 FOR T=1 TO 400::NEXT T::RETURN
```

16. Here are some interesting redefinitions for characters. To use them the proper format is "CALL CHAR(##,string) where "##" stands for the number of Character to be redefined and "string" is one of the following (or any other that you may have).

```
000804027F020408 = right arrow
00102040FE402010 = left arrow
081C2A4908080800 = up arrow
0008080B492AC108 = down arrow
00FF             = solid line
0000FE2828282828 = pi symbol
00083C4848483C08 = cent sign
0002020404482810 = check mark
```

17. The IMAGE statement (eg - 100 IMAGE ###.##) can be used with the DISPLAY AT statement using the following format -

```
DISPLAY AT(5,12):USING 100:A
```

18. Instead of using the IMAGE statement you can define a variable in the image you would like the output to look and then say "USING variable name".

```
eg - 100 F$="###.##"
     110 DISPLAY AT(12,1):USING F$:A
```

Of course, unlike the IMAGE statement which can be anywhere in the program, the variable would have to be defined BEFORE using it in a DISPLAY AT or PRINT statement.

19. When using the DISPLAY AT statement you can use TAB to properly locate where further information is to be displayed.
eg: To set up the following display

```
MAIN MENU:
  1 - Edit
  2 - Add
  3 - Exit
```

you can set up each line with an individual DISPLAY AT statement or you can do the following:

```
DISPLAY AT(5,5):"MAIN MENU";TAB(7);"1 - Edit";TAB(7);"2 -
Add";TAB(7);"3 - Exit"
```

This will put the information on 4 separate lines because when the computer tries to perform the TAB(7) it finds that that location has been already bypassed on the present row and therefore it automatically goes to the next row.

20. Did you know that you could delete a file when you close it. The statement is: CLOSE #1:DELETE

21. When programming in XB it pays in two ways to squeeze as many statements as you can into each program line. The first reason is that it saves memory by eliminating line numbers; the second is that it speeds up execution by eliminating the need for the program to process extra lines of code.

22. Another method to save memory by reducing the size of a program is to replace a constants used with a variable. This is assuming that that constant is used a number of different times in the program.

23. When you are editing a program and accidently erase a line by pressing FCTN 3 to get the line back simply type in a single quote mark and then press ENTER. This gives a syntax error and the erased line is back because the change was not syntactically correct and thus not acceptable. The putting in of the quote mark must be done before moving from the line that was erased.

24. If you need to know if CALL INIT has already been executed in your program put in the following lines:

```
10 CALL PEEK(8198,A,B)::IF A=170  AND B=85 THEN xxx ELSE CALL
   INIT          !!xxx = line number to go to if CALL INIT
                         already executed.
```

25. Here's a use of the MIN and MAX statements:

MIN - If a variable is restricted to being no higher than 6 you would normally say IF A>6 THEN A=6 however you can say A=MIN(A,6)

MAX - If a variable is restricted to being no lower than 6 you would normally say IF A<6 THEN A=6 however you can say A=MAX(A,6)

26. You can LIST a program to disk by stating LIST "DSKn.program". This gives a D/V 80 file which is then readable by TI-Writer. This can be helpful for putting program listings in documents and allows you to use RS to amend or FS to locate something in a long program.

27. Here's a short program to write DATA lines which can then be merged into another program.

```
100 ON WARNING NEXT
110 DISPLAY AT(10,1)ERASE ALL:"ENTER FIRST LINE NUMBER:"::
    ACCEPT AT(10,25)BEEP VALIDATE(DIGIT)SIZE(4):LN
120 DISPLAY AT(12,1):"ENTER INCREMENT"::
    ACCEPT AT(12,17)BEEP SIZE(3)VALIDATE(DIGIT):I
130 DISPLAY AT(14,1):"ENTER FILENAME:"::
    ACCEPT AT(14,16)BEEP VALIDATE(UALPHA,DIGIT)SIZE(10):FN$
140 OPEN #1:"DSK1."&FN$,VARIABLE 163
150 DISPLAY AT(2,6)ERASE ALL:"PRESS ENTER TO END"::
    DISPLAY AT(22,1):"ENTER A LINE OF DATA:"::
    LINPUT D$
160 IF D$="" THEN 190
170 PRINT #1:CHR$(INT(LN/256)&CHR$(LN-256*INT(LN/256))
       &CHR$(147)&D$&CHR$(0)
180 LN=LN+I::
    GOTO 150
190 PRINT #1:CHR$(255)&CHR$(255)
200 CLOSE #1::
    END
```

This will save your DATA lines in a Merge format almost ready to be merged into you program. Before this can be done you must do the following:

i) type NEW and press ENTER to clear memory
ii) MERGE in the saved DATA lines. (ie - MERGE DSK1.filename
iii) EDIT each DATA line by retyping (typing over) the word DATA
iv) SAVE the edited DATA lines in the MERGE format (ie - SAVE DSK1.filename, MERGE)

It is now ready to be put into you program.

THIS ARTICLE HAS BEEN PUT TOGETHER FROM MANY SOURCES
BY BILL SPONCHIA.

==

TI BITS Number 23
By Jim Swedlow

[This article originally appeared in the User Group of Orange County, California ROM]

A DIRTY DOZEN
Being a compendium of things small and not so small that are worth a word or two.

DOCS, Part 1
On disk documentation is a good way to tell you how to use a program. There are a number of ways to print them. The program may print the docs. You may have to print them through the Formatter. Other times there is a special program just to print the docs.
It would be nice if there was a README file that told you how to print the docs.

TI WRITER, Part 1
Sometimes, when you display a disk text (DV80) file on your screen, the last line looks like hieroglyphics. When you save a TI Writer file using SaveFile, the very last record contains the tab and margin settings. The characters are outside the ASCII 32 to 127 range, so they show on your screen as strange graphics.
When you save a file with PrintFile, the tabs and margins are not saved as TI Writer thinks that you are going to send your file to a printer. PrintFile, however, will accept any legal devise name (PIO, DSKn.FILENAME, RS232, etc).

FAIRWARE, Part 1
People who use Fairware but don't send the author a contribution earn a place in the dirty dozen.

NEWSLETTERS, Part 1
If you ever read through all the various TI news letters that TI User Groups publish you will be awed by the vast array of information that they purvey. From original software to hardware fixes to reviews to commentary, the range is immense.
Why is this in the dirty dozen?^ Because you probably have never looked at all of these wonderful resources.

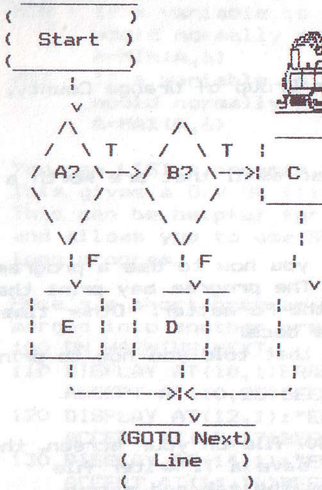
THIS AND THAT, Part 1
Another category in the dirty dozen is presentations at User Group meetings that are so complex, so techy that no one understands what is being said. Drives members away.

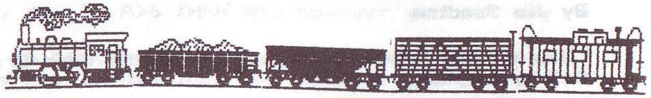
FAIRWARE, Part 2
Right next to users who do not contribute are Fairware authors who receive our contributions but fail to acknowledge them. It is good manners to say thanks. Just taking the money discourages future support.

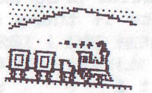
THIS AND THAT, Part 2
An IF^THEN problem that many of us have is figuring out how to match IF's and ELSE's.
Reading from the start of the line, each ELSE matches the last unmatched IF. Consider the following:

```
IF A THEN IF B THEN C ELSE D ELSE E
```

This can be displayed as a flow chart- see next page--->







DOCS, Part 2
It would also help if programmers had a novice user or two read their docs to make sure they tell you how to use the program.

NEWSLETTERS, Part 2
If you do get around to reading through old newsletters you will see articles that run something like this:
"Your officers are getting sick and tired of doing everything ourselves. If some of you don't start helping, things just aren't going to get done".
Folks, this does not solve the problem. Broad band appeals in newsletter articles almost always result in nothing. The only way to get folks to help is to ask them, one to one.

THIS AND THAT, Part 3
The PE Box fuses get special attention in the dirty dozen. TI put a 1.25 amp slow blow fuse outside the PE Box (the one you can get to). Inside the box (actually inside the main transformer) they put a one amp regular fuse. Guess which fuse goes when you get a short?

TI WRITER, Part 2
When FIll and ADjust are on, TI Writer always adds two spaces after a period. This is good for a sentence but bad for names and abbreviations. Mr. Jones looks wrong.
There is an easy fix. Use the circumflex (^) as a required space. If you type this in the Editor:
  Mr.^Jones
You will get this from the Formatter:
  Mr. Jones
Makes a better final document.

CLOSING SHOT
One last item to fill this dirty dozen. The thing in the TI World that bothers me the most is when folks loose sight of our prime objective. The 4A will have strong support as long as we hang together and keep our eye on the ball. For us, the center issue is survival.

Enjoy.

```
X   X   BBBB      # 18
 X X    B   B
  X     BBBB      By
 X X    B   B     Jim
X   X   BBBB      Swedlow
```

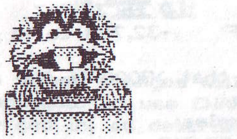
[This article originally appeared in the User Group of Orange County, California ROM]

WHAT IS A NIBBLE, ANYWAY?
This time I am going to try and explain all of the various number words we run across. With luck, after you finish reading this, you will have some understanding of bit, byte, nibble, word, hex, binary and where -31952 really is in memory. With luck.

Computers really think in binary. In this numbering system there are two numbers, 0 and 1 (or, if you are a computer, off and on). While this works for your 4A, binary is cumbersome for humans. For example, in binary 41,576 is 1010001100011100.

Hex, or hexadecimal, has sixteen numbers from zero to F. Here are the first sixteen numbers in binary, decimal and hex:

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111



The next number would be 16 or >10 or b1000 (> means hex and b means binary).
One binary number is a bit. Four bits is a nibble. So, 10 or A or 1010 takes four bits or a nibble to express.

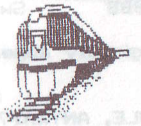
A byte is eight bits or two nibbles. With a bit you can count from zero to one. A nibble gets you from zero to fifteen. The range of byte is:

Base	Low	High
Binary	0	11111111
Hex	0	FF
Decimal	0	255

You have probably noticed the numbers 16 and 255 when using your TI. ASCII character run from 0 to 255. There are sixteen colors (1 to 16, really 0 to 15). A string can be up to 255 characters long. And on and on.

Before tackling the next thing, a word, lets see if we can decode something. Lets take b10100 or >14. To convert either number to decimal, we need a method:

```
  >14 is >10 plus >4
  >10 is 16 and >4 is 4
  16 plus 4 is 20
  Hence, >14 is 20
```



```
  b10100 is b10000 plus b100
  b10000 is 16 and b100 is 4
  16 plus 4 is 20
  b10100 is 20
```

Further than that I cannot go in this space.

A word is sixteen bits or four nibbles or two bytes. The range of a word is:

Base	Low	High
Binary	0	1111111111111111
Hex	0	FFFF
Decimal	0	65,535

But there are no negative numbers. Since we need them, we use something called twos compliment (which is way beyond the scope of this column and this writer). I can tell you, however, the impact;

Hex range	Decimal Range
0-7FFF	0 to 32,767
8000-FFFF	-32,768 to -1

Remember that >8000 is the next number after >7FFF.

Some examples:

7FFF is	32,767
8000 is	-32,768
FFFF is	-1
0 is	0

Confused? So was I until I worked with it for a while. These conversion rules may help:

>>Any number less than or equal to 32,767 requires no conversion.

>>Subtract 65536 from any number over 32,767.

>>Add 65536 to any number less than zero.

This conversion process can be expressed in basic:

```
AD=AD+65536*(AD>32767)
```

If AD is the address, this returns the same number if AD is less than or equal to 32767. If AD is greater than 32767, the test returns true (-1) and a negative 65536 is added to AD. Try it on your computer.

Bottom line time. Suppose you see CALL PEEK(-31952,A,B). Where is -31952? Well, since it is less than zero, we add 65536 and get 33584 or >8330. Now you know!

Enjoy!

TIPS FROM THE TIGERCUB
No. 60
1 June 1990

My stock of Tigercub Software catalogs is depleted and it would not pay me to reprint it. Therefore I have released all copyrighted Tigercub programs, except the Nuts & Bolts Disks, for free distribution providing that no price or copying fee is charged. All of my Tigercub programs have been added to my TI-PD library and are cataloged, by category, in Supplement #8.

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to $5.00. If I run out of printed documentatiion, it will be supplied on disk.

My TI-PD library now consists of 400 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename, Basic programs converted to XBasic, etc. The price is just $1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #2 with Supplement #8, listing all titles and authors, is currently available for $1 which is deductible from the first purchase.

Here are a couple of improvements to the CHARFIX subprogram published in Tips #58.

```
  29000 SUB CHARFIX(HX$()):: D
 ISPLAY AT(12,1)ERASE ALL BEE
 P:"Transliterate punctuation
 ?" :: ACCEPT AT(12,28)SIZE(1
 )VALIDATE("YN"):Q$ :: IF Q$=
 "N" THEN 29004
 29007 CALL CHARVIEW(HX$())
 29009 SUB CHARVIEW(HX$())
```

And call the routine by CALL CHARFIX(HX$()). These changes will avoid unwanted transliteration, and will make it possible to use CHARFIX for ASCII 24-31 and 144-159, if BXB has been merged in, as described in Tips #55.

The Spring 1990 issue of the TI*MES newsletter from England contained an interesting challenge - write a program in any language to find the lowest power of 7 which contains six sevens in succession, i.e. "777777". The computer cannot solve this by any normal means, because it soon goes into scientific notation in which large numbers are rounded off into long strings of zeros. So, I taught it to multiply the old-fashioned way -

```
100 A$=STR$(7):: Y=1
110 Y=Y+1 :: FOR J=LEN(A$)TO
 1 STEP -1 :: E=(VAL(SEG$(A$
,J,1))*7+X)/10
120 X=INT(E):: F=(E-X)*10 ::
 X$=STR$(F)&X$ :: NEXT J
130 IF X>0 THEN X$=STR$(X)&X
$
```

```
140 IF POS(X$,"777777",1)<>0
 THEN 160
150 A$=X$ :: X$="" :: X=0 ::
 GOTO 110
160 PRINT "7^";STR$(Y);"=";X
$
170 PRINT #2:"7^";STR$(Y);"=
";X$
```

The answer? 7^175=78011207 91220815810240464127911180777
77771881820069326361118396985716038858440266717799156064
7169989331265664440734763224 8554716494939953912586437943

My TI-99/4A computed that in 24 minutes. Would someone like to try it on the 9640?

[Please refer to TI*MES #29 for a faster way to do it sjs]

Anyway, I thought I would use the same method to solve precise multiplication of numbers too large to be computed directly. This routine will multiply two numbers of up to 28 digits each, and will handle decimals and negative numbers. For even larger numbers, change the ACCEPTs to INPUTs and if necessary change the DIM.
The only limitation seems to be that the result cannot contain more than 256 digits and even that could be programmed around.

```
100 DIM C$(100)
110 DISPLAY AT(12,1)ERASE AL
L:"FIRST NUMBER?" :: ACCEPT
AT(14,1)VALIDATE(NUMERIC)BEE
P:A$
120 IF SEG$(A$,1,1)="-" THEN
 A$=SEG$(A$,2,255):: M=1
130 A=LEN(A$):: D1=POS(A$,".
",1):: IF D1>0 THEN A$=SEG$(
A$,1,D1-1)&SEG$(A$,D1+1,255)
:: D1=A-D1
140 DISPLAY AT(16,1)ERASE AL
L:"SECOND NUMBER?" :: ACCEPT
 AT(18,1)VALIDATE(NUMERIC)BE
EP:B$
150 IF SEG$(B$,1,1)="-" THEN
 B$=SEG$(B$,2,255):: M=M+1
160 Y=LEN(B$):: D2=POS(B$,".
",1):: IF D2<>0 THEN B$=SEG$
(B$,1,D2-1)&SEG$(B$,D2+1,255
):: D2=Y-D2 :: D1=D1+D2 :: Y
=Y-1
170 FOR J=Y TO 1 STEP -1 ::
W=W+1 :: B=VAL(SEG$(B$,J,1))
:: FOR K=LEN(A$)TO 1 STEP -1
 :: A=VAL(SEG$(A$,K,1))
180 D=(A*B+X)/10
190 E=INT(D):: F=(D-E)*10 ::
 C$(J)=STR$(F)&C$(J):: X=E :
: NEXT K
200 IF X>0 THEN C$(J)=STR$(X
)&C$(J)
210 C$(J)=C$(J)&RPT$("0",W-1
)
220 X=0 :: NEXT J
230 L=LEN(C$(1)):: FOR J=1 T
O Y :: L2=LEN(C$(J)):: IF L2
<L THEN C$(J)=RPT$("0",L-L2)
&C$(J)
240 NEXT J
250 FOR J=LEN(C$(1))TO 1 STE
P -1 :: FOR K=1 TO Y :: G=G+
VAL(SEG$(C$(K),J,1)):: NEXT
K
260 G=(G+H)/10 :: L=INT(G)::
 G=(G-L)*10 :: D$=STR$(G)&D$
 :: H=L :: G=0 :: NEXT J
270 IF H>0 THEN D$=STR$(H)&D
$
280 IF D1>0 THEN D$=SEG$(D$,
1,LEN(D$)-D1)&"."&SEG$(D$,LE
N(D$)-D1+1,255)
290 IF M=1 THEN D$="-"&D$
300 PRINT D$
```

And this one will add up an almost unlimited number of integers of almost any length - I haven't figured out how to get it to line up decimals.

```
100 CALL CLEAR :: DIM C$(100
)
110 DISPLAY AT(12,1):"Input
from D":"  (D)isk or":"  (K)
eyboard?" :: ACCEPT AT(12,12
)VALIDATE("DK")SIZE(-1):Q$ :
: IF Q$="K" THEN 140
120 DISPLAY AT(12,1)ERASE AL
L:"Filename? DSK" :: ACCEPT
AT(12,14):F$ :: OPEN #1:"DSK
"&F$,INPUT
130 X=X+1 :: LINPUT #1:C$(X)
:: M=MAX(M,LEN(C$(X)):: IF E
OF(1)<>1 THEN 130 ELSE CLOSE
 #1 :: GOTO 160
140 DISPLAY AT(12,1):"Press
ENTER when finished":"":""
150 X=X+1 :: INPUT C$(X):: M
=MAX(M,LEN(C$(X))):: IF C$(X
)<>"" THEN 150 ELSE X=X-1
160 FOR J=1 TO X :: IF LEN(C
$(J))<M THEN C$(J)=RPT$("0",
M-LEN(C$(J)))&C$(J)
170 NEXT J :: FOR J=M TO 1 S
TEP -1 :: FOR K=1 TO X :: G=
G+VAL(SEG$(C$(K),J,1)):: NEX
T K
180 G=(G+H)/10 :: L=INT(G)::
 G=(G-L)*10 :: D$=STR$(G)&D$
 :: H=L :: G=0 :: NEXT J
190 IF H>0 THEN D$=STR$(H)&D
$
200 PRINT D$
```

It is easy to invert characters on the screen simply by making the foreground "on" pixels a lighter color than the background "off" pixels - but when you make a screen dump, you will find that the "on" pixels will print and the "off" pixels will not.
Key this in, SAVE it by SAVE DSK1.INVERSE,MERGE and then merge it into any program by MERGE DSK1.INVERSE, call it at any point by CALL INVERSE(A,B), (A and B are the first and last ASCII to be inverted), and you will have all "on" pixels turned off and vice versa.

```
31111 SUB INVERSE(A,B):: FOR
 CH=A TO B :: CALL CHARPAT(C
H,CH$)
31112 FOR J=1 TO 16 :: CH2$=
CH2$&SEG$("FEDCBA9876543210"
,POS("0123456789ABCDEF",SEG$
(CH$,J,1),1),1):: NEXT J ::
CALL CHAR(CH,CH2$):: CH2$=""
 :: NEXT CH
31113 SUBEND
```

Here is a truly remarkable discovery by Bill Hudson of the Central Ohio Ninety Niners. This 2-line program will allow you to RUN a variable name such as - A$="DSK1.PROGRAM"
You can write lines before these, after these, and even RES the program. You can also use MOVE from GK UTILITY. You can do anything to the program you want as long as you don't change the content of line 1000. The line number does not even have to be 1000 BUT IT MUST BE THE FIRST LINE THAT YOU KEY IN!! You can merge a program into this but can't merge this into a program. Line 900 can also be a different line number but program execution must go to that line first.

```
900 FOR Z=1 TO LEN(A$):: CAL
L LOAD(-41+Z,ASC(SEG$(A$,Z,1
)),0):: NEXT Z :: CALL LOAD(
-41,LEN(A$)):: CALL LOAD(-44
,4+LEN(A$))
1000 RUN "DSKx.1234567890"
```

It's been a long time since we had a screen dis- play to watch just for the fun of it, so here is a tinygram -

```
100 CALL CLEAR :: FOR SET=1
TO 14 :: CALL COLOR(SET,SET+
1,SET+2):: NEXT SET :: CALL
SCREEN(2):: CALL VCHAR(1,1,3
1,768)
110 FOR CH=32 TO 136 STEP 8
:: CALL CHAR(CH,"FF000000000
000FF"):: NEXT CH
120 X=INT(RND*6+1)*2-1 :: Y=
INT(14*RND+1)*8+32 :: FOR R=
12-X TO 12-INT(RND*X):: CALL
 HCHAR(R,5,Y,R)
130 CALL HCHAR(25-R,5,Y,R)
140 CALL HCHAR(R,28-R,Y,R)
150 CALL HCHAR(25-R,28-R,Y,R
)
160 ON INT(2*RND+1)GOTO 170,
190
170 CALL HCHAR(R,4+R,Y+8,25-
R*2)
180 CALL HCHAR(25-R,4+R,Y+8,
25-R*2)
190 NEXT R :: GOTO 120
```

This is a challenging and educational math puzzler which I think is unlike anything you have seen. I had it in my Tigercub catalog for 7 years and sold just 18 copies. If you don't want to key it in, it is now one of the programs on TI-PD disk No. 1300.1.

```
100 GOTO 140
110 J,K,ST,LV,I,R(),T,X,A,A$
,X$,B,B$,C,C$,D,D$,AY,BY,B@$
,BY$,CY,CY$,C@$,Q,Y(),Y@,X@(
),FLAG,R$,RL,Z,YY,D@(),Q$
120 CALL CLEAR :: CALL CHAR
:: CALL COLOR :: CALL VCHAR
:: CALL SCREEN :: CALL KEY :
: CALL SOUND
130 !@P-
140 CALL CLEAR :: FOR J=1 TO
 12 :: CALL COLOR(J,5,16)::
NEXT J
150 CALL VCHAR(1,3,32,672)::
 DISPLAY AT(5,1):" @$%#*#+#
RITHMATIK #+#%$@ "
160 DISPLAY AT(10,1):" Selec
t difficulty level -": :" Ty
pe 1 or 2"
170 CALL KEY(0,K,ST):: IF ST
<1 THEN 170
180 IF (K<49)+(K>50)THEN 170
190 LV=K-48
200 CALL VCHAR(1,3,32,672)::
 FOR I=1 TO 4 :: RANDOMIZE
210 R(I)=INT(RND*10):: IF R(
I)=0 THEN 210
220 FOR T=1 TO I-1 :: IF R(I
)=R(T)THEN 210
230 NEXT T
240 NEXT I :: X=R(1)*1000+R(
2)*100+R(3)*10+R(4)
250 A=INT(4*RND)+1
260 ON A GOSUB 330,340,350,3
60 :: A$=X$
270 B=INT(4*RND)+1 :: IF B=A
 THEN 270
280 IF (LV=1)*(LEN(STR$(R(B)
/R(A)-INT(R(B)/R(A))))>2)THE
N 250
290 ON B GOSUB 330,340,350,3
60 :: B$=X$
300 C=INT(4*RND)+1 :: IF C=A
 THEN 300
310 IF C=B THEN 300
320 ON C GOSUB 330,340,350,3
60 :: C$=X$ :: D=10-A-B-C ::
 ON D GOSUB 330,340,350,360
:: D$=X$ :: GOTO 370
330 X$=" 1st " :: RETURN
340 X$=" 2nd " :: RETURN
350 X$=" 3rd " :: RETURN
360 X$=" 4th " :: RETURN
370 AY=R(B)/R(A):: BY=ABS(R(
C)-R(B)^2):: IF BY=0 THEN 38
0 ELSE 390
380 B@$="" :: BY$=" equal to
" :: GOTO 400
390 B@$=STR$(BY):: BY$=" mor
e or less than"
400 CY=ABS(R(D)-R(C)-R(B)-R(
A)):: IF CY=0 THEN 410 ELSE
420
410 CY$=" equal to" :: C@$="
" :: GOTO 430
420 CY$=" more or less than"
 :: C@$=STR$(CY)
430 DISPLAY AT(2,1):" I have
 a 4-digit number ":" with n
o two digits the":" same." :
: DISPLAY AT(6,1):" The";B$;
"digit is";AY;" times the";A
$;"digit."
440 DISPLAY AT(9,1):" The";C
$;"digit is ";B@$;BY$;" the
square of the";B$;" digit."
:: DISPLAY AT(14,1):" The";D
$;"digit is ";C@$;" ";CY$;"
the sum of the other digits"
450 DISPLAY AT(18,1):" What
is the number?" :: ACCEPT AT
(20,2)VALIDATE(DIGIT)SIZE(4)
BEEP:Q :: IF Q=X THEN 530
460 Y(1)=INT(Q/1000):: Y(2)=
INT((Q-1000*Y(1))/100):: Y(3
)=INT((Q/100-INT(Q/100))*10)
:: Y(4)=(Q/10-INT(Q/10))*10
:: IF Y(B)<>INT(Y(A)*AY)THEN
 570
470 IF BY<>0 THEN 490
480 IF Y(C)<>Y(B)^2 THEN 570
 ELSE 500
490 IF (Y(C)<>Y(B)^2+BY)*(Y(
C)<>Y(B)^2-BY)THEN 570
500 IF CY<>0 THEN 520
510 IF Y(D)<>Y(A)+Y(B)+Y(C)T
HEN 570 ELSE 530
520 IF (Y(D)<>Y(A)+Y(B)+Y(C)
+CY)*(Y(D)<>Y(A)+Y(B)+Y(C)-C
Y)THEN 570
530 DISPLAY AT(22,1):" Corre
ct!": : :: FOR J=1 TO 2 :: C
ALL SOUND(100,392,5):: CALL
SOUND(100,440,5):: CALL SOUN
D(100,494,5):: CALL SOUND(10
0,523,5)
540 NEXT J :: CALL SOUND(100
0,523,5,392,5,330,5)
550 DISPLAY AT(24,1):" Hit a
ny key"
560 CALL KEY(0,K,ST):: IF ST
<1 THEN 560 ELSE 200
570 DISPLAY AT(22,1):" Wrong
." :: CALL SOUND(900,30000,3
0,30000,30,400,30,-4,0):: DI
SPLAY AT(23,1):" Type A to t
ry again or Z":" to see the
number"
580 CALL KEY(0,K,ST):: IF ST
<1 THEN 580
590 IF K=65 THEN 450
600 IF K=90 THEN 610 ELSE 58
0
610 DISPLAY AT(22,1):" The n
umber was";X:" " :: GOTO 550
:: END
```

Nearly out of memory and all out of ideas. More next time, maybe.
Jim Peterson
Tigercub

MACHINE CODE TUTORIAL
Part Four

FILE HANDLING

File specifications describing a file's record length, length format, data and file format, and mode of operation are accumulated in a block of memory known as the Peripheral Access Block (PAB). The writing,reading and updating of file data is handled by resident routines called Device Service Routines (DSR's). For every file you must open, read or write, and close it.

You must first define the file characteristics. The actual number of bytes in the PAB are variable. PAB byte zero instructs the DSR which operation you wish to perform.(e.g. open, read, write, close).

You cannot use opcode >08 (scratch) with disks. The number one byte defines the files open mode, record type, type of data, and sequential or random. Bits 0,1, and 2 are used to report error conditions as they occur.

Bytes 2,3 (1 word) contain the address in VDP RAM to be used as a buffer as the data is read or written. Byte 4 defines the logical record length in bytes. For variable length records this value is the maximum length. The largest value is >FF (255). Byte 5 defines the number of bytes to be written for a write operation, or the actual number of bytes to be read for a read operation.
For fixed length records PAB byte 4 and 5 should be set equal when writing and will always be equal when reading. For variable length records PAB byte 5 can be tested to determine the actual record length on a read and can be dynamically changed for each write. PAB byte 5 can never exceed byte 4.
PAB bytes 6 and 7 are only used for relative records. This word contains the relative record number to access. The MSB is ignored so the range of values is 0 to 32,767. Byte 8 is only used for files to be stored on cassette tape. Set byte 8 to >60 (screen offset) for cassette access. Byte 9 is the file descriptor length. Byte 10 is the file description.

PABs are coded in your program and then placed in VDP RAM with VMBW. The first free address in VDP RAM for PABs is usually >F80. VDP RAM free space extends through >37D6.
Actual access is obtained by the DSR routine. REF DSRLNK must be included in your program. The pointer needed by DSRLNK is the address of the file descriptor byte. This value must be placed at >8356. DSRLNK is then envoked with a BLWP instruction. When errors occur the equal bit of the status byte is set.

You must design the layout of each file.

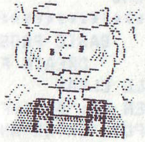
Recommend you read the following references in your editor assembler manual:

Section 16.2.2 page 251
Section 16.2.4 page 262
Section 16.5 page 270 through Section 16.5.4 page 271

EDITORIAL

The only way to become a proficient programmer is to write programs. For extra practice try converting this BASIC program to assembler. You have had all the programming pieces in previous tutorials, you just have to put them together. All you lack is confidence.

```
100  CALL CLEAR
200  OPEN #1:"PIO",SEQUENTIAL,
DISPLAY, OUTPUT, VARIABLE 80
300  REM SUBSTITUTE YOUR PRINTER
NAME FOR "PIO"
400  FOR I=1 TO 10
500  PRINT "YOUR NAME"
600  PRINT #1:"YOUR NAME"
700  NEXT I
800  CLOSE #1
900  END
```



```
****************************************
*  THIS ROUTINE DEMOS FILE ACCESS       *
*   BY RALPH MOLESWORTH                 *
****************************************

       DEF   START
       REF   DSRLNK,VSBW,VMBW,VSBR,VMBR
STATUS EQU   >837C
POINTR EQU   >8356             DSR POINTER ADDRESS
BUFADR EQU   >1000             VDP RAM ADDR FOR REC BUFFER
PABADR EQU   >F80              VDP PAM ADDR FOR PAB
READ   BYTE >02                "READ" OP-CODE
CLOSE  BYTE >01                "CLOSE" OP-CODE
EOF    DATA 0                  EOF FLAG
PAB    DATA >0014,BUFADR,>5000,>0000,>000A  PAB DATA
       TEXT 'DSK1.FILE1'
ERRMSG TEXT 'I/O ERROR='  DSR ERROR MESSAGE
CPUBUF BSS   80                CPU RAM REC BUFFER ADDR
FNAME  EQU   CPUBUF            FIRST NAME ADDR
LNAME  EQU   CPUBUF+14         LAST NAME ADDR
LEN    BSS   2                 ACTUAL RECORD LEN WORKSPACE
RETURN BSS   2                 SAVE RTN ADDR AREA
WR     BSS   >20               WORKSPACE REGISTERS
START  MOV   R11,@RETURN  SAVE RETURN ADDR
       LWPI WR
       LI    R0,PABADR         VDP RAM ADDR FOR PAB
       LI    R1,PAB            CPU RAM ADDR FOR PAB DATA
       LI    R2,20             LENGTH OF DATA
       BLWP @VMBW              WRITE PAB TO VDP RAM
       BL    @DSR              OPEN THE FILE
       MOVB @READ,R1           LOAD READ OPCODE INTO R1
       LI    R0,PABADR         LOAD PAB ADDR INTO R0
       BLWP @VSBW
       CLR   R4                R4 IS RECORD COUNTER
READF  BL    @DSR              PERFORM DSR ROUTINE
       MOV   @EOF,@EOF         CHECK FOR EOF
       JNE   EOJ
       INC   R4                ADD 1 TO REC COUNT
       CI    R4,3              CHECK FOR THIRD REC
       JNE   READF             IF NOT THIRD, READ AGAIN
                               * ----> continued ---->
```

```
        LI   R0,PABADR+5  ADDRESS OF THE CHARACTER COUNT
        BLWP @VSBR        READ COUNT INTO LEFT BYTE R1
        SRL  R1,8
        MOV  R1,R2        MOVE VALUE TO R2
        MOV  R1,@LEN      SAVE VALUE IN LEN
        LI   R0,BUFADR    VDP RAM REC BUFFER ADDR
        LI   R1,CPUBUF    CPU RAM ADDR FOR REC
        BLWP @VMBR
        LI   R0,290       PUT UP FIRST NAME
        LI   R1,FNAME
        LI   R2,14
        BLWP @VMBW
        LI   R0,305       PUT UP LAST NAME
        LI   R1,LNAME
        MOV  @LEN,R2      MOV RECORD LEN TO R2
        AI   R2,-14       SUBTRACT LEN OF FIRST NAME
        BLWP @VMBW
        JMP  EOJ          GOTO END OF JOB
DSR     LI   R6,PABADR+9  LOAD R6 WITH DESCRIPTOR LEN
        MOV  R6,@POINTR   MOV ADDR TO POINTER
        BLWP @DSRLNK
        DATA 8            DATA NEEDED BY DSR LINK
        JEQ  DSRERR       CHECK FOR ERROR
        RT
DSRERR  INC  @EOF         SET EOF INDICATOR
        LI   R0,PABADR+1  ADDRESS OF PAB BYTE 1
        BLWP @VSBR
        SRL  R1,13        SHIFT HIGH ORDER 3 BITS TO LOW
        CI   R1,5         CHECK FOR EOF VAL=5
        JNE  IOERR        IF NOT EOF THEN OTHER ERROR
        RT
IOERR   AI   R1,>30       MASK ERROR CODE
        SLA  R1,8         SWAP BYTES
        LI   R0,299       PUT UP ERROR CODE
        BLWP @VSBW
        LI   R0,288       DISPLAY ERROR MSG
        LI   R1,ERRMSG
        LI   R2,10
        BLWP @VMBW
EOJ     MOV  @EOF,@EOF    IF EOF REACHED, DSR WILL
        JNE  NOCLOS       CLOSE FILE
        MOVB @CLOSE,R1    MOVE CLOSE OP-CODE TO R1
        LI   R0,PABADR    LOAD PAB ADDR
        BLWP @VSBW        WRITE CLOSE OP-CODE TO PAB 0
        BL   @DSR         CLOSE FILE
NOCLOS  DECT @RETURN      ALTER RETURN ADDR
        MOV  @RETURN,R11
        RT
        END
```

```
***************************************************
*     BASIC PROGRAM TO                            *
*         CREATE FILES                            *
*100 CALL CLEAR                                   *
*110 OPEN #2:"DSK1.FILE1",OUTPUT,VARIBLE 80       *
*120 INPUT "ENTER AN E WHEN DONE":X$              *
*130 IF X$="E" THEN 190                           *
*140 INPUT "FIRST NAME?":FN$                      *
*150 IF LEN(FN$)>14 THEN 140                      *
*160 INPUT "LAST NAME?":LN$                       *
*170 PRINT #2:FN$,LN$                             *
*180 GOTO 120                                     *
*190 CLOSE #2                                     *
*200 END                                          *
***************************************************
     =====  end =====
```

This program is really silly, but it is short, and I wonder if you can see how it gets the answer it does... the math is all honest!

```
100 DISPLAY AT(1,1)ERASE ALL:"NUMBER OF MONTH(1-12)"
110 ACCEPT AT(2,12)SIZE(2)VALIDATE(DIGIT):A :: IF A<1 OR A>12 THEN 110
120 DISPLAY AT(3,1):A;"x 4=";A*4 :: A=A*4
130 DISPLAY AT(4,1):A;"+13=";A+13 :: A=A+13
140 DISPLAY AT(5,1):A;"x 25=";A*25 :: A=A*25
150 DISPLAY AT(6,1):A;"-200=";A-200 :: A=A-200
160 DISPLAY AT(8,1):"Input date (1-31):" :: ACCEPT AT(8,19)SIZE(2)VALIDATE(DIGIT
):B :: IF B<1 OR B>31 THEN 160
170 DISPLAY AT(10,1):A;"+";B;"=";A+B :: A=A+B
180 DISPLAY AT(11,1):A;"x 2=";A*2 :: A=A*2
190 DISPLAY AT(12,1):A;"-40=";A-40 :: A=A-40
200 DISPLAY AT(13,1):A;"x 50=";A*50 :: A=A*50
210 DISPLAY AT(15,1):"Input last two digits of     year eg 91:"
220 ACCEPT AT(16,16)SIZE(2)VALIDATE(DIGIT):B
230 DISPLAY AT(18,1):A;"+";B;"=";A+B :: A=A+B
240 DISPLAY AT(19,1):A;"-10500=";A-10500 :: A=A-10500
250 DISPLAY AT(24,1):"ANY KEY FOR ANOTHER"
260 CALL KEY(5,A,B)
270 IF B<1 THEN 260
280 RUN
290 END
```

==

This is also a rather silly short program. Can you guess what will happen when you type run I wonder? This only goes to prove that you can put together a math formula for almost anything....

```
100 ! R LODGE / S SHAW
110 ! jolly program
120 CALL CLEAR
130 FOR A=1 TO 2
140 FOR B=1 TO 4
150 X=2-ABS(SGN(B-3))
160 FOR C=1 TO X
170 PRINT CHR$(84-7*A+5*B-8*X);
180 NEXT C
190 NEXT B
200 PRINT CHR$(A+31)
210 NEXT A
220 END
230 ! there is an equation for everything!
```

Short enough for all of you to type in, AND in TI Basic too.
