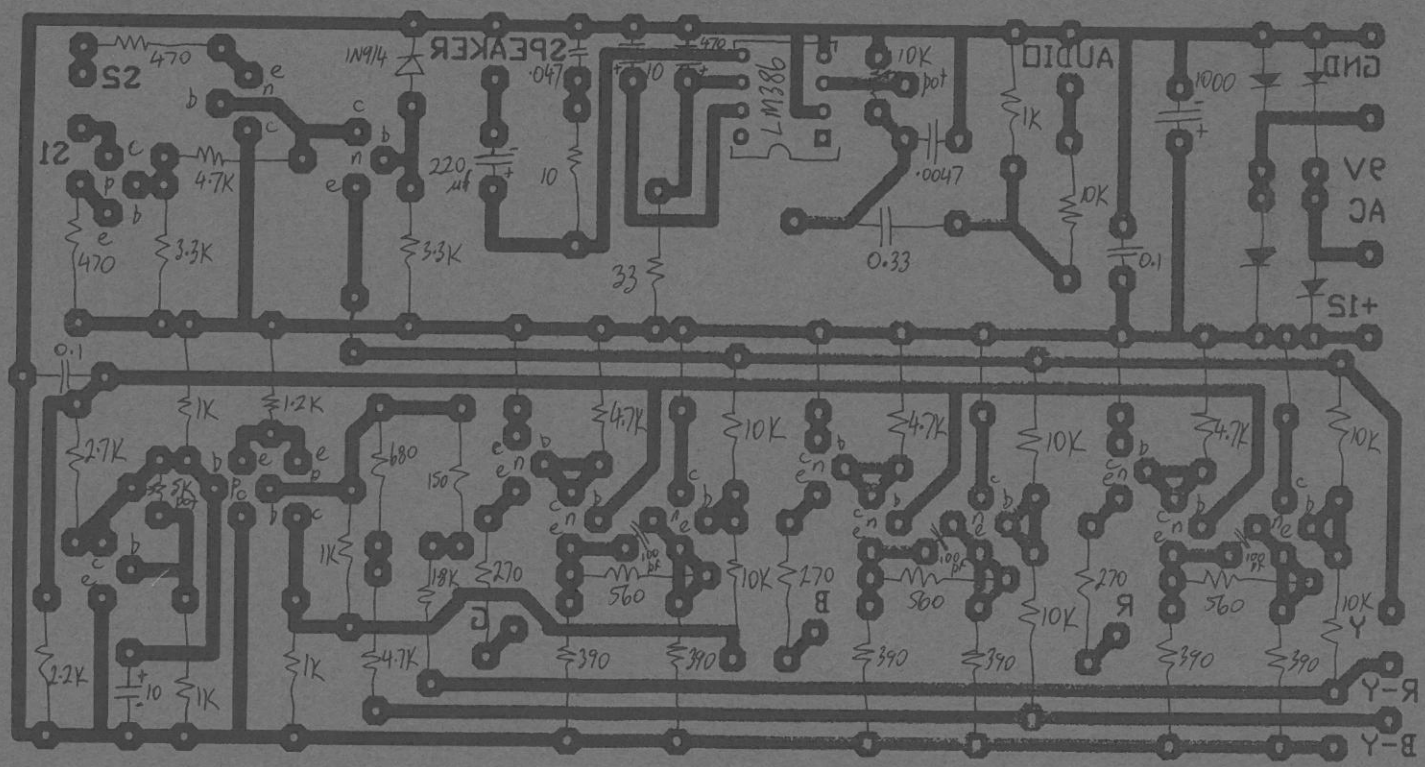
TISHUG NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 11, Number 7 August, 1992

Registered by Australia Post - Publication No. NBH5933

Overlay for new RGB monitor interface PCB



MONINT Bottom Layer

Sydney, New South Wales, Australia $3

TIsHUG (Australia) Ltd
A.C.N. 003 374 383

TIsHUG News Digest

August 1992

All correspondence to:

P.O. Box 1089
Strawberry Hills, NSW 2012
Australia

The Board

Co-ordinator
Dick Warburton (02) 918 8132
Secretary
Terry Phillips (02) 797 6313
Treasurer
Geoff Trott (042) 29 6629
Directors
Rolf Schreiber (042) 85 5519
Russell Welham (043) 92 4000

Sub-committees

News Digest Editor
Bob Relyea (046) 57 1253
BBS Sysop
Ross Mudie (02) 456 2122
BBS telephone number (02) 456 4606
Merchandising
Percy Harrison (02) 808 3181
Publications Library
Russell Welham (043) 92 4000
Software library
Rolf Schreiber (042) 85 5519
Technical co-ordinator
Geoff Trott (042) 29 6629
TI-Faire co-ordinator
Dick Warburton (02) 918 8132

Regional Group Contacts

Central Coast
Russell Welham (043) 92 4000
Coffs Harbour
Kevin Cox (066) 53 2649
Glebe
Mike Slattery (02) 692 8162
Hunter Valley
Geoff Phillips (049) 42 8176
Illawarra
Geoff Trott (042) 29 6629
Liverpool
Larry Saunders (02) 644 7377
Northern Suburbs
Dennis Norman (02) 452 3920
Sutherland
Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues	$35.00
Associate membership	$10.00
Overseas Airmail Dues	A$65.00
Overseas Surface Mail Dues	A$50.00

TIsHUG Sydney Meeting

The August Meeting will start at 2.00 pm on 1st of August at Ryde Infant School, Tucker Street, Ryde. The Assembly Class will run from 10.00 am to 1.00 pm for those wishing to attend.

Printed by
The University of Wollongong
Printery Services.

TIsHUG News Digest ISSN 0819-1984

Index

Thank you to Tony Bell for his re-typing of articles which appear in this month's TND

All articles appearing in this month's issue of the TND are available as text files on disk ready for the formatter. Newsletter editors please note that, if you wish to re-print any articles, contact us, stating which articles you are interested in and giving the date of the TND. These will be dispatched to you promptly at the cost of the media plus postage.

Editor's Comment

by Bob Relyea

Some of the things that interested us a lot in the early days of the computer have not received much attention lately, such as the Speech Synthesizer. I saw an interesting article in one of the overseas magazines about using the Terminal Emulator II for speech and found it to be simple but interesting. I retyped it and it will probably appear in this month's edition. It is fun sometimes to get out some of the gear we used ages ago and refresh our minds on how it worked. I never spent much time on the synthesizer but I plan on using that article as a basis of really looking into it. By playing around with all parameters you can even get various accents, such as Scottish or Australian (the original words already have an American accent). Have a 'go' at it.

Secretary's Notebook

by Terry Phillips

An unseasonably warm day for the July meeting saw quite a number of members in attendance. This was the second buy, swap, sell meeting for the year and again there seemed very little on the tables, although Peter Schubert's offerings, particularly the bargain priced box of boards and other bits and pieces, were proving popular. I do not know how others feel but it would appear that most members now have all the equipment etc. that they want and/or are reluctant to sell off anything. Maybe it is all being kept for spare parts. Should the buy, swap, sell formats be continued? Any comments would be welcome.

At the last Directors' meeting, the upcoming November Faire was one of the topics discussed. It seemed to me that the prospect of holding a "Faire Dinner" on the Saturday night would be a good idea. As the Faire is being held in Ashfield and, as I am a local resident of that suburb, it was suggested that I should make some enquiries and see what is available.

Now Ashfield, over the past few years, has become rather cosmopolitan, particularly in the number and variety of restaurants and clubs available for patronage and most of these are within a few minutes walk of the Faire venue. What I would like is a bit of feedback as to what type of venue would suit the majority who would like to attend a dinner. Available are Chinese, Korean, Thai, Italian and Polish - the majority are licensed but a couple do offer BYO - and prices would generally be in the $15 to $20 a head range. At the bottom end of the spectrum is Western Suburbs Leagues Club which offers a $2 meal throughout the day. Under 18's are welcome in the dining area of the club if accompanied by an adult. Give it a bit of thought and let me know if you are likely to attend. An idea of numbers is required to make the appropriate booking.

Colin Christensen, a stalwart in the Brisbane Group, and a member of our own group, has sent me down one of his hardware items, CADET CONSOLE EXPANDER, and it will be demonstrated at the August meeting. The idea of this nicely put together gadget is to provide a cheap method of expansion into word processing for the console only owner. It provides 3 additional CALLS - WP for word processor, FO for formatter and GL for the game loader. It would also be useful for those who sometimes need to set up a second TI99/4A for a quick bit of word processing, or game playing, but find it difficult to get dad, mum, son or daughter off the main system. Price suggested by Colin is $100. Come along and see it at the August meeting. Colin also confirmed he will be a definite visitor to the Faire.

Doug Moller, a member from Aitkenvale, North Queensland, has written to advise that he recently rescued a swag of computer books that were destined for the local dump. They are mostly for 'other' computers, ZX Spectrum, VIC20, Apple etc. but if someone wants them they are free. Doug is planning a trip south and says he hopefully will be in the Sydney area during late November. He would like to meet other members during his motoring trip south, so if you would like to meet him let him know by writing to him at PO Box 719 Aitkenvale, QLD 4814.

Doug also presented a problem that hopefully some one may be able to answer. If you can please write to him at the above address. This is his problem:
"I recently purchased a second hand printer with centronics interface. I have an AT card. Before I connect it to the TI99/4A, I would like to ensure that connections are all OK, ie, no problems with cable connections. Also I think it is due for a new ribbon and I would like to know the original manufacturer type number to facilitate ordering. The printer itself is a Microbee DP1000 friction or sprocket feed, 100cps, 1064NUB."

That's all for this month. See you at the August meeting. ○

Computers in Education

by Jim Peterson, Tigercub Software, USA

Comments found in a recent issue of School Tech News under the above heading were published in the Spirit of 99 October 1991 issue. I guess the article makes a lot of sense and the comments are equally valid in this country.

Are we spending too much money on computers in School? Sometimes we are, says Michael Wessells, Professor of Phsychology at Randolph-Macon College in Ashland, Virginia.

Wessells thinks computers are used too often for "electronic page-turning" - presenting lengthy texts which could be handled just as well in book form - and for displaying "electronic flashcards" in learning languages. He thinks the money spent on such seemingly misguided use of computers in the classroom might be used more wisely by allocating costly computer time for "richer, more satisfying, more interactive activities that use the power of the computer" to better advantage. Wessells says he is neither skeptical nor "starry-eyed" where computers are concerned. He sees his job as "encouraging people to ask hard questions" about computers, especially computers in schools.

Computers have a significant place in education and a great contribution to make there, Wessells says. But as with other technologies, such as educational television, "naive idealism" tends to make some people "see computers as the salvation of education". He points out "I don't think they are".

Setting priorities for computer education is important in this age of fast-evolving technologies, Wessells says. What goal should schools aim for to make every student capable of using a state of-the-art computer? Programming is less important in the curriculum now, but schools still struggle with the issue of what to teach about computers, he adds.

Another aspect of computer education that concerns Wessells is "the social context of computers in education", as he puts it. Consider gender inequities, for example. At early ages (up to third grade or so), boys and girls use computers about equally. Then a divergence appears; boys start using computers more, while girls use computers less. This divergence reflects the emphasis and expectations of a culture that treats computers as a masculine domain, Wessells says.

Class inequities are a problem too, Wessells goes on. Using sixth graders as a case in point, Wessells says that computer use at school is proportional to computer use at home - so that children from middle class and upper class households enter the computing environment at school with an advantage over lower class children who have less opportunity to use computers at home.

Schools in areas where lower class children predominate, find that students use computers less frequently and in less productive ways than in schools serving more affluent communities. This is one way computers "can be used to reinforce the dominant social order", Wessells says. ○

continued from page 18

19. Again rewind to TOF. Now set the paper to print just under the top images. Be SURE to do power Off/On.

20. Do the BREAK RUN thing again. this time it is not necessary to select an image, you may go directly to Options. Select Text (only). Select New or Old as you choose, just like printing a normal label or letterhead. When ready, select Print, then Head from the menu, accept the default 27 as to the column to Put it query. Enter only 1 to the How Many query until you see if you like it. You will be returned to the menu, to do more if you want.

continued on page 4

TIsHUG Shop with Percy Harrison

Well last months TND was certainly aimed at Ramdisk owners with Geoff Trott, Rolf Schreiber and myself all having something to say about ways and means of trying to prevent Ramdisk crashes. Fortunately we all seemed to be in agreement as to what measures to take so hopefully the frequency of crashes from now on will not be so prevalent, I say this with tongue in cheek and fingers crossed.

Sales of the Wang coloured monitors have been very good and at the time of writing this I only have one fixed base unit left in stock. I currently have a problem supplying the interface kit as I have run out of printed circuit boards and am waiting for Geoff to get a new batch made.

I have just added a 3.5 drive to my system so if any member would like the club freeware programs supplied on 3.5 disks please advise me of your requirements and I will run them off for you; unfortunately they will be slightly dearer than our programs on 5.25 disks to cover the higher price of the blank disk i.e. $2.50 as compared with $2.00. Because of the anticipated small demand for this size of disk the programs will not be available unless they are ordered from me in advance.

Lastly we have been successful in purchasing another batch of 5.25 half height drives (black face) which are available at the same price as our previous drives i.e. $65.00 each.

Club Software Disks

A119 J P Drawing Vers 3.0 DSSD $2.00
A145 Adventures (Scott Adams) DSSD $2.00
A212 G Language SSSD $2.00
A214 Plus Vers 1.0 SSSD $2.00
A245 Telco Vers 2.3 DSSD $2.00
A261 Assembly Language Games SSSD $2.00
A380 Super-Cataloger SSSD $2.00
A382 Boot (40 Column Vers) SSSD $2.00
A386 Boot (Hard Disk Vers) SSSD $2.00
A401 Pix Version 1.2 SSSD $2.00
A411 Miner/49er, Espial SSSD $2.00
A420 Atari Games #1 SSSD $2.00
A430 Configuring Funnelweb SSSD $2.00
A431 Object Linker Vers 3.0 SSSD $2.00
A436 Hotbug SSSD $2.00
A437 Nasty and Segregation SSSD $2.00
A438 More Assembly Games SSSD $2.00
A439 Multiplan Exercises DSSD $2.00
A448 Tips Vers 1.7 SSSD $2.00
A448A Tips Graphics #1 SSSD $2.00
A448B Grips (Tips Companion) SSSD $2.00
A450 Funnelweb 4.40 DSDD $2.00
A450A Funnelweb 4.40 (3 Disks) SSSD $4.00
A451 Multiplan Vers 4.02 SSSD $2.00
A453 The Nutcracker Suite SSSD $2.00
A456 Remembrance Disk (Music) SSSD $2.00
A457 Anna Magdalene Music SSSD $2.00
A462 Rediskit SSSD $2.00
A463 TI-Exam SSSD $2.00
A464 Il Pastor Fido Vivaldi DSSD $2.00
A465 Lute Music SSSD $2.00
A466 Best of DOM #5 DSSD $2.00
A467 The Singing TI SSSD $2.00
A468 Speech #1 SSSD $2.00
A472 TI Writer Supplement SSSD $2.00
A473 DM 1000 Version 5.0 SSSD $2.00
A474 GIF Pictures (80 column card) DSDD $2.00
A481 Artconvert DSSD $2.00
A481A Artconvert (2 Disk Set) SSSD $3.00
A482 Horizon Utilities SSSD $2.00
A483 TI Tiler SSSD $2.00
A484 Mac-labels SSSD $2.00
A485 YEO SSSD $2.00
A486 Genealogy Record Keeping DSSD $2.00
A487 XHI (80 column card) DSDD $2.00
A488 Utilities LA 99ers SSSD $2.00
A489 Fontart #1 SSSD $.....$2.00
A490 Fontart #2 and #3 DSSD $2.00
A491 Designing Graphic Screens SSSD $2.00
A492 Microkey SSSD $2.00
A493 Disk Manager 99 Vers 2.0 SSSD $2.00

TCC1 Tigercub Collection #1 SSSD $2.00
TCC2 Tigercub Collection #2 SSSD $2.00
TCC3 Tigercub Collection #3 SSSD $2.00
TCC4 Tigercub Collection #4 SSSD $2.00
TCC5 Tigercub Collection #5 SSSD $2.00
TCC6 Tigercub Collection #6 SSSD $2.00
TCC7 Tigercub Collection #7 SSSD $2.00
TCC8 Tigercub Collection #8 SSSD $2.00

TC911 Display Calculator SSSD $2.00
TC1015 Word Processing Utilities SSSD $2.00
TC1122 Screen Fonts-Peterson DSDD $2.00
TC1131 Gemini Printer Utilities SSSD $2.00
TC1145 Telecommunications SSSD $2.00
TC1210 Graphics Printing SSSD $2.00
TC1211 TI Artist Pictures #1 SSSD $2.00
TC1212 TI Artist Pictures #2 SSSD $2.00
TC1213 TI Artist Pictures #3 SSSD $2.00
TC1219 R Kazmer's Xmas Card SSSD $2.00
TC1220-1229 Tips (10 Disks) DSSD $20.00

Packaging and postage charges:

	Surface	Airmail
1 to 2 Disks	$1.90	1.90
3 to 9 Disks	$2.90	$3.60
10 to 20 Disks	$3.90	$4.80
TI Artist Plus	$3.00	$3.70
Display Master	$3.00	$3.70
TI Base	$3.00	$3.70
TI Sort	$3.00	$3.70
5.25 inch half-height drive (1.25 Kg)	refer to your local post office	

Bye for now.

o

Assembly Class

by Ross Mudie

The assembly class on the 4th of July was attended by six members. The topic was saving data in memory image format using DSRLNK under Extended Basic. The next assembly class will be at Ryde Infants school preceding the August meeting on Saturday the 1st of August. The subject will carry on from the July class and cover loading a memory image data file. All members are welcome to attend, the planned start time for the class is 10am.

Members planning to attend this class should read pages 262 and 291 to 304 of the Editor Assembler manual before attending the class. It is essential for class members to bring their own Editor Assembler manual to this class. o

Treasurer's Report

by Geoff Trott

I have received some encouraging comments about my articles on C. I will try and keep them up but I would like some help in knowing the direction that I should go. I was going to try and solve a problem for David Peters and his printer to allow him to print out graphics from a variety of programs but this will only be of immediate interest to him. However, the writing of the program should provide me with examples useful to you all, I hope.

Income for June	$2561.90
Payment for June	$2832.15
Excess of expenses over income for June	$270.25 o

TIsHUG Software Column

by Rolf Schreiber

Software Releases for August

DISK A476 is another disk from the Hamilton User Group who were based in Ontario Canada. The disk is menu driven and contains a collection of XB and assembly utilities and games.

Filename	Size	Type / Length	
AIRWOLF	16	PROGRAM	3750
AIRWOLFS3	28	DIS/VAR	80
AIRWOLFSM	2	DIS/VAR	80
BOOGENS2	21	PROGRAM	4879
CALL/PEEK	7	DIS/VAR	80
COMPOSER	24	PROGRAM	5789
GAME/9	20	PROGRAM	4609
JOYST_EXB	4	PROGRAM	676
LOAD	16	PROGRAM	3598
LOADER	10	PROGRAM	2129
QUEENBEE	7	PROGRAM	1472
QUES	25	DIS/VAR	80
SC/CARD	7	PROGRAM	1328
TJOY	15	PROGRAM	3534
TJOY;A	16	DIS/VAR	80
TJOY;C	4	DIS/VAR	80
TJOY;O	20	DIS/FIX	80
UNSTACKER	5	PROGRAM	902
YLOAD	10	DIS/FIX	80

DISK A494 is a collection of miscellaneous programs on the first of ten disks from Earl Raguse, who has also been contributing TI Forth articles in the TND.

Filename	Size	Type / Length	
2COLUMDOCS	5	DIS/VAR	80
2COLUMNS	9	PROGRAM	1850
BASSNOTES	3	PROGRAM	485
CAT	5	PROGRAM	973
COMPRESS	2	PROGRAM	197
CURSOR	3	PROGRAM	460
DIR	8	PROGRAM	1705
EARLSLABEL	7	PROGRAM	1523
EXTRACTDOC	3	PROGRAM	501
FONT	8	PROGRAM	1746
FORMAT	3	DIS/VAR	80
HORIZON	8	PROGRAM	1725
ILOVEYOU	3	PROGRAM	397
INVRSVIDEO	3	PROGRAM	415
LEFTMARGIN	3	PROGRAM	365
LINEFEED	3	PROGRAM	432
LIST	4	PROGRAM	661
LOAD	2	PROGRAM	46
LOWSOUNDS	3	PROGRAM	323
MARGINS	5	PROGRAM	965
MEDLEY	32	PROGRAM	7892
MONEY	69	INT/VAR	254
NEWCOLOR	4	PROGRAM	634
NIM	22	PROGRAM	5207
NOISE	4	PROGRAM	683
P1	27	DIS/VAR	80
PTEST	2	PROGRAM	108
SIGN	5	PROGRAM	944
SUBEXTRACT	4	DIS/VAR	163
WAVESBIRDS	4	PROGRAM	616
WORMY	7	PROGRAM	1414

DISK A504 is a very good disk cataloguer from Ron Rutledge. Full documentation comes on the disk.

Filename	Size	Type / Length	
CATALOG	4	PROGRAM	711
DIRECT/DOC	43	DIS/VAR	80
DIRECTOR	42	PROGRAM	10244
DIRLAB/DOC	16	DIS/VAR	80
DISK/LABEL	17	PROGRAM	3929
LOAD	4	PROGRAM	730
PRINT/DOC	3	PROGRAM	388

Tigercub Software

TCC-9 is a collection of miscellaneous games which can be selected from a menu which autoloads from Extended BASIC. The following items are on the disk:

1 Nervous Breakdown
2 Jelly Beans
3 Memory Sharpener
4 Quick 'N' Think, Simon Says
5 Bingle
6 Air Rescue
7 Air Defense
8 Air Traffic Controller
9 Arrow Zap
10 Attacker
11 Bat Attack
12 Bonkers
13 Adventure with Dracula
14 Egg Wars

TC-820 is a disk on Health and the Human Body. The following programs can be selected from the menu:

1 Biorhythms and You
2 Biorhythm Compatibility
3 Biorhythm #1
4 Biorhythm #2
5 Biorhythm #3
6 Biorhythm #4
7 Calorie Cop
8 Daily Nutrition
9 Diet Management
10 Diet Rite
11 Name That Bone

TC-860 is called Astronomy #1. The following programs can be selected from the menu:

1 Astronomy
2 14 Constellations
3 57 Constellations
4 Elliptical Orbits
5 Jupiter's Moons
6 Phases of the Moon
7 Parabolic Orbits
8 Planetarium
9 Planets
10 Skyscape

continued from page 2

21. This time select Head, 27 (if you liked that position, else adjust), then enter the number to print. I recommend you print less than a million. The ribbon tends to fade. When done, tear off the paper, separate the sheets, and write letters, letters, letters. I always tear off the tractor holes and use my single sheet feeder.

continued from page 21

```
440 GOTO 460
450 RIGHT=RIGHT+1
460 PRINT "IF YOU SCORED:"
470 PRINT
480 PRINT "0 - VERY POOR"
490 PRINT "1 - POOR"
500 PRINT "2 - GOOD"
510 PRINT "3 - VERY GOOD"
520 PRINT "YOU GOT ";RIGHT;" RIGHT AND ";WRONG;" WRONG"
530 END
```

continued from page 18

```
   REPLACE HTEMP WITH "  Dear " | FN
     PRINT HTEMP,(LF)
  SELECT 2
  TOP
  PRINT ALL LINE
   PRINT (FF)
RETURN Copyright Martin A. Smoley 1990
*
* Prints the Letter part of RENEW with
* a Letterhead.
```

Dijit AVPC 80 Column Card

Review by Jan Alexandersson, Sweden

1. GENERAL

I have installed an 80-column card from DIJIT with 192 kbytes RAM in the PE-box. It has an EPROM with 50 Hz PAL as default but I can change it to 60 Hz NTSC in a program. The card is delivered with 50 or 60 Hz EPROM dependent of which country you live in. The card comes with an 8 page manual which describes how you connect the card and modify the console. You will also get 3 disks with program and text files. There comes no help for a programmer with the card. If you want to write your own programs which uses the special features of the video processor then you must have a manual from Yamaha. You can buy the manual from Asgard.

The same 9938 video processor is used in Myarc 9640 Geneve and Mechatronic 80 column card. This means that many programs for these may be used with the DIJIT AVPC card. MSX-2 and Nintendo (DIJIT and Asgard have different opinions about this) also have the same 9938 video processor but they have another CPU than the TI-99/4A.

You can order the AVPC from DIJIT Systems, 4345 Hortensia St., San Diego, CA 92103, USA. The price is USD 250 with 192 kbytes RAM plus USD 10 air mail to Europe. You can also buy modified EPROM from DIJIT for the RS232 card from TI(USD 22), Myarc(USD 20) or Corcomp (USD 20 for 24 or 28 pins).

2. MODIFY THE CONSOLE

You must open the console and modify the mother board. This is well described in the manual. Usually you do not need to solder but I had to use the solder iron to get the two metallic screens apart from the mother board. You modify by cutting one trace on the card. The video processor 9929A (PAL in Europe) must be lifted up and one pin shall be bent out. You shall fasten a ready-made cable on this and also at two other places on the card. All this was easy to do. The most difficult thing was to press down the video processor which has a lot of pins. Look very carefully that no pins are bent and put the chip down very slowly. The pins will spread apart somewhat so you must press them together. You have the same problem if you want to change EPROM in a unit like the RS232-card. The whole chip has a white cream which lead away the heat. It is important that this will remain and you may buy more of that.

3. RGB MONITOR

DIJIT recommends an RGB monitor with 12 MHz bandwidth and 0.42 mm CRT-pitch. You cannot use an ordinary TV-set with SCART for RGB because it is only good for 40 column display. I have a Phillips

CM 8833 monitor with SCART contact for RGB. I had to make a cable with SCART and 6-pin DIN. My cable looks like this:

AVPC	Monitor
6-pin DIN	SCART contact
1 Blue	7
2 Red	15
3 Ground	5, 9 ,13
4 Green	11
5 Csync	20
6 +5 V	-

Notice that pin 20 Csync is only mentioned in the English part of the manual and not in the part for Sweden or any other language. I had to press the button for RGB-status on the back of the monitor to get a picture on the screen with AVPC. This was not necessary with TI-99/4A with RGB-modulator. I have not connected pin 16 (blanking) of the SCART-contact. It may have the same function to connect it to +5 V as RGB-status but as it works anyway I will not try it. DIJIT does not recommend the use of the blanking pin.

The sound must be connected to the old 6-pins DIN-contact of the console. I made the cable like this:

99/4A PAL	Monitor
6-pin DIN	SCART contact
5 Audio	2, 6 Stereo
6 Ground	4

Check very carefully that you have made the cable correct. Use a magnifying glass so you can see the pin numbers which is engraved in the plastic around the contact. Use an ohm meter to verify the connections.

There is also an output for composite video that you can use with a high resolution monochrome monitor. You cannot use a colour monitor because the colours will not look nice.

Phillips has up to now sold two types of monitors CM8833 (0.42 mm, 12 MHz) and CM8873 (0.31 mm, 18 MHz) but is now changing to 8CM852 (0.39 mm, 14 MHz) and 8CM875 (0.31 mm, 30 MHz). You must always get linear RGB, 15.6 kHz horizontal frequency, 50 and 60 Hz vertical frequency, Audio input. The AVPC has 15.75 kHz horizontal frequency but it will also work on a monitor for 15.6 kHz. You should ask if the monitor works with an Amiga before you buy it. You must always make the cable yourself. Phillips is sold in the USA as Magnavox.

4. NEWS AND PROBLEMS (EPROM version 1)

The AVPC with EPROM version 1 reserves 16 bytes high up in VDP which results in that SIZE in Extended BASIC will show 11824 bytes in VDP-RAM. The card uses CRU >1400 which means that you cannot have any other card at the same address. You must have an other card at CRU >1100 like a disk controller. The power-up will not work if this is missing. The AVPC will after its own power-up do the power-up at CRU >1100. I do not know in which order the other cards are searched and if CRU >1000 will be done. I do not think that this is any problem but it can be useful to know that a 32 kbytes EM is not enough. I hope there is no other card that takes over the power-up in the same way because two such cards could block each other.

A good news is that QUIT must be done with three buttons simultaneously, CTRL/FCTN/=. This will not work with a RAVE keyboard which only can sense two buttons.

DIJIT says that there may be programs that will not work with AVPC. All programs I have used has worked satisfactory. The picture is much more clear also in the usual graphic modes. In BASIC you will see some coloured or black stripes on the screen when an ERROR occurs. This problem is not seen in Extended Basic. You cannot activate the card by setting CRU >1400 with Easy Bug in Mini Memory because the AVPC works with interrupt. MG Diskassembler can despite of this disassemble the AVPC.

TI-FORTH must be changed so screen 54 row 11 becomes ... 07 4 VWTR. Graphic2 will not work otherwise. TURBO-PASC'99 must be changed so that sector >54 byte >25 becomes >54 (>50 before). The USA-version is correct but if you have bought it from Germany then you may have to change.

5. PROGRAMS FOR AVPC

The three floppy disks have many useful programs and texts:

- Funnelweb for 80 columns
- 11 different graphic demo programs
- Fractal
- Program for showing pictures from Geneve
- 7 pictures for Myarc Geneve
- Modified ROS for Horizon
- CALL LINK for mouse with Extended Basic

- Interrupt for mouse with sprite 1
- DSR routine for TI-Artist with mouse
- SC-DOS for RAM >6000
- Modified TI-FORTH for 80 columns
- Cabling of ATARI Trackball
- Cabling of mouse and light pen
- Questions and answers about AVPC

There are also other programs which uses the new features in the 80 column card. TELCO for data communication also works with 80 column. This is very helpful because many data bases is made for 80 column computers. TELCO with EPROM version 1 will show garbage characters on the screen but besides this it works. TELCO uses also AVPC as a RAM disk so different program modules in TELCO can be stored in AVPC RAM. The AVPC can hold up to 27 modules compared to 3 modules with an ordinary 99/4A.

Many programs which is made for Myarc Geneve may be used by the AVPC-card. This is the case for most programs in Geneve GPL-mode. I have used the program PALETTE from Micropendium february 1989 which works fine with my AVPC. The program gives you the possibility to test all 512 colours in the common graphic mode which is used in Extended Basic. The colour is defined by setting a value of 0-7 for all the three basic colours red, green and blue. Black will be 0-0-0 and white 7-7-7. Dark yellow will be 6-6-1. You can change all the 16 colours numbered 1-16 in Basic.

6. GRAPHIC

The 80 column card with its 9938 video processor has 10 different graphic modes compared to 99/4A which has only the first four in the table below:

Mode	pixels	chars	colours	sprite mode	memory kbytes
Multicolour	64x48	-	16	1	4
Graphic1	256x192	256	16	1	4
Graphic2	256x192	768	16	1	16
Text1	256x192	256	2	-	4
Text2	512x212	256	4	-	8
Graphic3	256x192	768	16	2	16
Graphic4	256x212	bitmap	16	2	32
Graphic5	512x212	bitmap	4	2	32
Graphic6	512x212	bitmap	16	2	64
Graphic7	256x212	bitmap	256	2	64

You can always choose among the 512 different colours for all modes (G7 has only 256 colours) but you cannot use more than the table shows at the same time. You can have twice as many pixels vertically with interlace so you can have maximum 512x424 pixels. The picture is not stable on my monitor with interlace but rumors say that it will be better with a multisync monitor. Notice that Graphic4-7 have true bitmap so every pixel can have its own colour which is not the case with Graphic2. With Text2 you get 80 columns and 26.5 rows. Sprite model has single coloured sprites and maximum 4 sprites per pixel row. Sprite mode2 has multicoloured sprites and maximum 8 sprites per pixel row.

The 9938 has 24 control registers, 15 command registers, 16 colour registers and 10 status registers. The video processor has instructions to move memory VDP-VDP, CPU-VDP and VDP-CPU and can also draw lines and more. I have used VDP register 9 to change 26.5/24 rows (bit 0), Interlace (bit 4) and PAL50Hz/NTSC60Hz (bit 6). You can try this in Extended Basic with a small CALL LINK routine which sets VDP register. These three have the decimal value 128, 8 and 2. If you write 138 to VDP-REG 9 then you have set 26.5 rows, interlace and PAL 50 Hz. The value 2 to VDP-REG 9 gives 24 rows, no interlace and PAL 50 Hz. Notice that also NTSC 60 Hz can be used very well.

7. MOUSE

There are instructions for cabling of the following mice: Amiga mouse, Atari mouse, Logitec P-7 Bus Mouse, Mouse System M4, Microsoft Bus Mouse (requires pull-up resistors) and Myarc mouse. Apparantly there is no standard for a mouse cable so all these have different cables. I have a Commodore 1352 mouse with two buttons for an Amiga. I believe this is the easiest one to make a cable for because all shall be mirrored. This means that you draw the cable straight between two 9 pin DB and get it mirrored. Do not connect the mouse to AVPC without an adapter cable. Ground and +5V must be changed. AVPC has ground and +5V as Myarc Geneve but the other pins are different.

The video processor can sense two buttons on the mouse. Myarc Geneve mouse has three buttons because the CPU 9995 can also sense one button (No 1). A comparison gives this:

CALL LINK XB	DIJIT AVPC mouse	Myarc Geneve mouse	TI-ARTIST
1	1 left	3 right	fire
2	2 right	2 middle	space
3	1+2	2+3	-
4	-	1 left	-
5	-	1+3	-
6	-	1+2	-
7	-	1+2+3	-

In the demo program for the CALL LINK routine you can change for AVPC as follows:

```
line 160 ...  IF B=3 THEN ...
line 170 ...  IF B=1 THEN ...
```

The demo program for mouse that moves sprite 1 may be changed:

```
line 115 IF M2 AND M3 THEN ...
line 120 IF M2=1 THEN ...
line 122 IF M3=1 THEN ...
```

TI-Artist 2.01G has two DSR-routines for joystick, EXTDSR and JOYST. They are identical except that JOYST has an extra instruction CLR @RBAND at the beginning. This label RBAND is not mentioned in the manual. All DSR-routines for mouse or joystick may also use the arrow keys and ENTER.

8. RS232 CARD

There is a bug in all RS232 cards from TI, Myarc and Corcomp. If you will use terminal program for modem which uses interrupt then you must change the EPROM. There is one such program which does not use interrupt, OMEGA. I have changed EPROM on my Myarc RS232 card. This sits in a socket so it is easy to do it. If you have a TI RS232 this will be more difficult because it is soldered. You must cut the leads of the circuit (do not desolder) and then clear each hole with the soldering iron. You may then solder a new socket in which you mount the the new EPROM from DIJIT. If you have a TI card you must throw away the old PROM which can be saved for a Myarc card. Corcomp has also the EPROM in a socket from the beginning.

9. REFERENCES

Micropendium:
October '87: Mouse interface for Extended Basic
November '87: Mouse input routine for TI-Artist
June '88: A full screen FORTH editor (2 colours)
September '88: Myarc, DIJIT rely on Yamaha 9938
February '89: Use palette master to mix colours

continued on page 20

Programs

from Stephen Shaw, England

FAST TYPER...

```
1 REM SILLY PROG BY S SHAW MARCH 1991
2 ! did you see COMPUTER WARS-the film? It is said that
the star, who was required to type fast into a computer
3  ! could not type, so a program just like this one was
used to give a good effect!
4 ! now adjust it how you wish and show your friends how
fast you can type
5  !  at  end of text string program will just stop with
this listing but can be
modified to do anything you wish!
6 !
100 A$="This is how a non-typist can produce information
on screen quickly, without "
110 A$=A$&"having to look at what keys are being bashed!
Just bash keys and watch how  perfect  text  appears  no
matter what you press"
120 PRINT A$: : : : : :
130 CALL KEY(5,A,B):: IF B<1 THEN 130
140 C=C+1 :: PRINT SEG$(A$,C,1);:: IF C=LEN(A$)THEN 160
150 GOTO 130
160 GOTO 160
```

==========

```
1 ! ADDING FRACTIONS
2  !  result  is  not  reduced eg 12/16 would usually be
shown as 3/4
100 CALL CLEAR
110 DISPLAY AT(10,5):"--- + --- = ---"
120 ACCEPT AT(9,5)SIZE(3)VALIDATE(DIGIT):A
130 ACCEPT AT(11,5)VALIDATE(DIGIT)SIZE(3):B
140 ACCEPT AT(9,11)SIZE(3)VALIDATE(DIGIT):C
150 ACCEPT AT(11,11)SIZE(3)VALIDATE(DIGIT):D
160 GOSUB 230
170 DISPLAY AT(9,16):USING "####":N
180 DISPLAY AT(11,16):USING "####":L
190 DISPLAY AT(14,1):"ENTER KEY FOR ANOTHER"
200 DISPLAY AT(1,1):"NORMAL RESULT=";A/B+C/D
210 ACCEPT AT(24,12):A$ :: GOTO 100
220 STOP
230 FOR X=2 TO B*D
240 IF INT(X/B)<X/B THEN 260
250 IF INT(X/D)=X/D THEN 270
260 NEXT X
270 L=X
280 N=INT(L/B)*A+INT(L/D)*C
290 RETURN
```

Now write a program which will reduce the answer to a more normal format- and also one which can take away as well as add. This is not a trivial test! and I will list a suitable program later- see if you can do it first!

==========

'PI' GENERATOR

PI is the ratio of a circles radius (distance from centre to edge) and its circumference (distance all the way round). In Extended Basic we have a fixed variable PI we can use. In TI Basic we can use the function ATN to derive PI. In machine code or most other languages we are on our own! WE could approximate to 3.14, but if greater accuracy is required? This little program derives PI to as many places as our language can support, without using any trigonometry, and is quite remarkably fast!

You may note the use of AND 2, which is using AND as a "logical operator", not too well covered in the manual. In Extended Basic, type in:

FOR T=1 TO 20 :: PRINT T;T AND 2 :: NEXT T and inspect the result!!!

```
1 ! CALCULATE PI
2 ! QUICKLY AND ACCURATELY
3 ! from W F DOSSETT, AUSTIN, TEXAS, 25 feb 88
4 ! for TI99/4A S SHAW 3/91
5 ! from John Machin, 1706
7 ! n=7 for 13 digit accuracy of TI99/4A Basic
8 ! n>7 if language goes beyond 13 digits!
9 ! eg n=35 for 53 digits
10 !
100 N=7 :: K=-1
110 K=K+1
120 I=-((2 AND 2*K)-1)
130 U=U+4*I/((2*K+1)*5^(2*K+1))
140 V=V+I/((2*K+1)*239^(2*K+1))
150 IF K<=N THEN 110
160 W=U-V :: PI2=4*W
170 ! DONE
180 PRINT PI2,PI
190 PRINT (PI2-PI)*1000,! COMPARE  with on board
variable
200 PRINT (PI2-(4*ATN(1)))*1000 ! compare with TI
Basic's calculation
210 PRINT "   ";(PI2-3.14)*100;:: DISPLAY AT(24,2)
SIZE(4):"3.14"
! displays ALL the digits actually in the variable!!!=13
incl dec point.
220 PRINT " 3.14159265358979 -etc"
! more accurate actual value for comparison!
230 END
```

999 ! from THE REC NEWSLETTER
March 1988 Vol 3 #2

The program is not derived from the standard mathematical series, stretching off to infinity, but from an approximation published around 1706, a little before computers were around! This, courtesy of Mr Dossett, has yielded a rather fast program!

==========

PRIMES...

Continuing with math related topics, prime numbers- that is numbers which are divisible only by themselves and 1. Asked to write a program to list prime numbers I suspect most of us would start from 1 and then try dividing every number by every number to see if it was divisible...

The first listing is fairly straight forward, but uses a few tricks to speed things up. In particular, if we know that a number is NOT divisible by two, it cannot possibly be divisible by any higher power of two, such as 4- or indeed any multiple of two. So we can check just the odd numbers, and also not bother testing for divisors greater than the first power of a known prime.

```
100 ! FIRST 100 PRIMES -NEATLY-
110 ! from Loren Krienke via THE REC NEWSLETTER March
1988 Vol 3 #2
120 ! for TI99/4A S SHAW
130 L=100 ! number required
140 DIM P(100)! P(L)-holds found primes, used as
divisors
150 DIM S(100)! S(L)-squares of primes, reduces number
of divisions needed
160 P(1)=2 ! first prime
170 PRINT P(1);
180 T=1 ! initialise loop
190 FOR I=2 TO L ! test non-even numbers:
200 T=T+2
210 FOR J=1 TO I-1
220 CALL MOD(T,P(J),R):: IF R=0 THEN 200
230 IF T>S(J)THEN 235 ELSE 240
235 NEXT J
240 P(I)=T ! found prime, store it
250 S(I)=T*T ! GREATER SPEED
260 PRINT T;!PRIME
270 NEXT I ! LOOP
280 STOP
290 ! 18 JUly 1986
300 ! Loren Krienke
310 ! San Diego, Ca.
320 !
330 SUB MOD(A,B,C)! C=A MOD B becomes CALL MOD(A,B,C)
340 C=INT(A-B*INT(A/B))
350 SUBEND
```

A similar but very much more opaque program follows, which has no obvious connection with the above, but it produces the same output!!! Check out the speed comparison between the two programs. Any difference? You may also wish to try looking for a larger number of primes!

```
100 ! FIRST 100 PRIMES -QUICKLY-
110 ! Dr H B Phillips from THE REC NEWSLETTER March 1988
Vol 3 #2
120 DIM P(300),X(12)
130 A=0 :: B=1 :: D=0.5 :: E=180
140 M=100 :: L=3 :: F=0
150 ! increase M for more- also increase DIMs.
160 PRINT 2;:: C=B :: IF M=B THEN END
170 L=INT((M/C)*L+F):: N=L+L+B
180 FOR I=B TO INT((SQR(N)-B)*D):: PP=P(I)
190 IF PP=B THEN 230
200 IF PP=A THEN PP=I+I+B :: PRINT PP;:: P(I)=PP :: C=
C+B :: IF C=M THEN END
210 IF X(I)=A THEN X(I)=(PP*PP-B)*D
220 FOR J=X(I)TO L STEP PP :: P(J)=B :: NEXT J :: X(I)=J
230 NEXT I :: IF F=0 THEN S=I
240 FOR I=S TO L
250 IF P(I)=A THEN PP=I+I+B :: PRINT PP;:: P(I)=PP :: C
=C+B :: IF C=M THEN END
260 NEXT I :: F=(M-C)*L/E :: S=L+B
270 GOTO 170
```

And no, I have no idea how this program works! When faced with a problem, the programmer usually has more than one way of attacking it, and it also usually happens that by restating the problem a faster solution can be found.

==========

Now for a graphic program requiring The Missing Link.

```
100 ! RABBITS AND FOXES
110 ! Dolores Garcia,Spain
120 ! Dr M Ecker, PA, USA
130 ! S SHAW FOR TI99/4A       EX BAS+TML   APRIL 91
140 CALL LINK("CLEAR"):: A,B=5
150 CALL LINK("PRINT",A,B,"Rabbits and Foxes"):: A=A+12
:: B=5
160 CALL LINK("PRINT",A,B,"Please input the variables
requested below. Suggested values are given for a
'typical' response. Try others!")
170 CALL LINK("PRINT",70,12,"RABBIT BIRTH RATE:")
180 CALL LINK("PRINT",80,12,"RABBIT DEATH RATE (Excl
eaten ones!):")
190 CALL LINK("PRINT",100,12,"RATE OF EATEN RABBITS:")
200 CALL LINK("PRINT",110,12,"FOXES DEATH RATE:")
210 CALL LINK("PRINT",120,12,"FOXES BIRTH RATE:")
220 CALL LINK("INPUT",70,150,A,4,".04"):: IF A<.00001
THEN A=0.4
230 CALL LINK("INPUT",90,150,B,6,".00005"):: IF B<1E-9
THEN B=.0005
240 CALL LINK("INPUT",100,150,C,4,".002"):: IF C<.00001
THEN C=.002
250 CALL LINK("INPUT",110,150,D,4,".03"):: IF D<.00001
THEN D=.04
260 CALL LINK("INPUT",120,150,E,5,".0002"):: IF E<1E-8
THEN E=.0002
270 CALL LINK("PRINT",140,12,"RABBITS START AT:")
280 CALL LINK("PRINT",150,12,"FOXES START AT:")
290 CALL LINK("INPUT",140,120,R,4,"300")
300 CALL LINK("INPUT",150,120,F,4,"30")
310 CALL LINK("CLEAR")
320 P=F*R
330 R=(1+A-B*R)*R-C*P
340 F=(1-D)*F+E*P
350 P=F*R
360 IF ER>500 THEN CALL LINK(180,180,"-.-.-."):: GOTO
360
370 IF NOT(F>0 AND R>0 AND F<200 AND R<640)THEN ER=ER+1
:: GOTO 330
380 CALL LINK("PIXEL",2+4*F,R*.7+2):: CALL
LINK("PRINT",181,1,"RABBITS:"&STR$(IN T(R))&" ")
390 CALL LINK("PRINT",181,110,"FOXES:"&STR$(INT(F))&" ")
400 ER=0
410 GOTO 330
420 END
```

This is a tricky little program - for The Missing Link, in this format it is a demo program, but if you leave the worm invisible all the time (by leaving Pen Erase on all the time) then you have a nasty little trick - especially if you disable break and quit. This is the fun version...

```
100 A,B=10
110 CALL LINK("PRINT",A,B,"The Missing Link is a disk
based utility which allows the Extended Basic programmer
to make use of many ")
120 CALL LINK("PRINT",A,B,"enhanced commands, including
true bit map graphics, which in turn allows more
powerful PRINT commands, and of ")
130 CALL LINK("PRINT",A,B,"course, full access to
sprites.")
140 CALL LINK("PRINT",A,B,"This program demonstrates
graphics outside a small window which, worm fashion, eat
up the text! Enjoy..")
150 ! WORM
160 ! SCIENTIFIC AMERICAN DEC 87  by S L Wentworth
170 ! TI99/4A BY S SHAW 91
180 ! EXT BAS + THE MISSING LINK
190 DIM XC(25),YC(25)
200 XC(2),YC(2),XC(1),YC(1)=100
210 D=0
220 T=1
230 RANDOMIZE
240 CALL LINK("WINDOW",80,80,110,120,1)
250 CALL LINK("PRINT",7,7,"WORM     HOLE")
260 CALL LINK("REVWIN")
270 WT=T
280 CALL MOD(T,25,T):: T=T+1
290 CALL LINK("PE")
300 CALL LINK("CIRCLE",XC(T),YC(T),4,0)
310 CALL LINK("PD")
320 C=RND
330 IF C<.5 THEN D=D+.1745 ELSE D=D-.1745
340 X=XC(WT)
350 Y=YC(WT)
360 NX=X+4*COS(D)
370 NY=Y+4*SIN(D)
380 IF NX>189 THEN NX=NX-189
390 IF NY>240 THEN NY=NY-240
400 IF NX<2 THEN NX=NX+189
410 IF NY<1 THEN NY=NY+240
420 XC(T)=NX
430 YC(T)=NY
440 !
450 CALL LINK("CIRCLE",NX,NY,4,0)
460 CALL LINK("PD")
470 GOTO 270
480 SUB MOD(A,B,C)
490 C=INT(A-B*INT(A/B))
500 SUBEND
```

==========

This is the promised fractional plus and minus program. I will not tell you how long it took me to get the displays like that....

```
100 ! FRACTIONAL + AND -
110 ! R CALDWELL  JAN 91
120 ! FOR TI99/4A BY
130 ! S SHAW APR 91
140 DIM Q(102)
150 DISPLAY AT(1,1)ERASE ALL:"FRACTIONAL + & -"
160 DISPLAY AT(7,1):"--------"
170 DE,X=1
180 ACCEPT AT(6,2)VALIDATE(DIGIT,"+-")SIZE(4):N(X)
190 IF N(X)=0 THEN 310
200 ACCEPT AT(8,2)VALIDATE(DIGIT)SIZE(4):D(X)
210 IF D(X)=0 THEN 310
220 DP(X)=N(X):: DE=DE*D(X)
230 CALL HCHAR(6,1,32,12)
240 CALL HCHAR(8,1,32,12)
250 DISPLAY AT(5,1):"PLUS:"
260 DISPLAY AT(3,1):"ENTER 0 TO TOTAL"
270 X=X+1 :: DISPLAY AT(7,15):"ITEM";X :: DISPLAY AT(8,
15):"-MAX 10-"
280 IF X=11 THEN 310
290 GOTO 180
300 REM
```

continued on page 14

TI-Bits Number 18

by Jim Swedlow, CA USA

[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, or 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).

If you are saving a file you will edit again or will run through the Formatter, use SaveFile. If you are saving a file some one else will view on disk, use PrintFile.

FAIRWARE, Part 1

People who use Fairware but do not 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 technical 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.

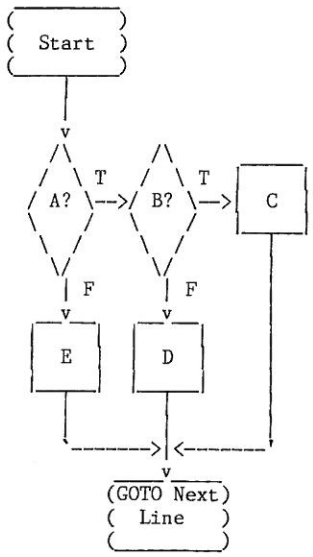
THIS AND THAT, Part 2

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:



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 do not start helping, things just are not 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?

The following description comes from Newt Armstrong. COM stands for the common tap of the primary side of the power supply transformer.

```
:.....................................:
:The primary has taps for 100, 120, 220:
:and 240 volt input.  The FAN is before:
:the INTernal FUSE between the 120V tap:
:       240---+ and COM.  The 120V tap:
:             ( is the center of the:
:       220---+ primary.              :
:             ( .....................:
: EXT         ( One fix is to rewire:
:  F          ( the COM and FAN leads:
:  U          ( to the 240V tap and re-:
+--S--+-120---+ gain transformer.  Taps:
|  E  |        ( for 100V and 220V will:
|     | 100---+ not work, but they are:
AC    |        ( not used anyhow.     :
IN   FAN   INT( .....................:
|     |     F ( The other fix is to dig:
|     |     U ( out the fuse carefully:
+-----+-COM-S-+ and replace it.  Do not:
:             E ( take the transformer:
:               apart.  Strip off the:
:insulation.  Be very, very careful.  :
:.....................................:
```

continued on page 10

Cataloguing Disks in XB

by Bob Relyea

Those of you, like me, who have an expanded system but do not have the standard TI disk controller card probably do not have access to an important booklet entitled 'DISK MEMORY SYSTEM' that accompanied the purchase of the TI controller. I was not aware of its contents until a recent regional meeting of the Illawarra group. I merely asked how to go about cataloguing a disk in an Extended Basic program (I am still adding to my Grade Standardising program!). Somebody said, "Oh, that is all explained in the TI Disk Memory System booklet", and handed one to me to borrow for a month or so. I finally got around to looking at it and upon reading through it coupled with a usual question or two to poor 'ol Ross Mudie, I had success. I am now working on a method that will result in printing out the catalog once displayed on the screen. That may be the subject of another article, however. In this article I am merely going to give you the contents of a couple of pages of the booklet that I found useful. The following is taken from pages 37 and 41 of the abovementioned booklet:

CATALOGUING FILES

By accessing the file index on a diskette, you can use TI Basic (or Extended Basic) to read a catalog of disk contents. The diskette-index (or catalog) file is an unnamed, INTERNAL-format, FIXED-length file. The following is an example of an OPEN statement that accesses the catalog on drive one:

```
100 OPEN#1:"DSK1."INPUT,RELATIVE,INTERNAL
```

NOTE: File-name is omitted, since the catalog is an unnamed file.

Every record in the catalog file contains four items: one string and three numeric values, written in INTERNAL format. There are exactly 128 records in the file, numbered from 0 through 127.

Record #0 contains information about the diskette. The string (up to 10 characters in length) indicates the name of the diskette, and the numerical items give the following information:

* the record type (always a zero for record #0)
* the total number of sectors on the diskette
* the number of available sectors on the diskette

Records 1 through 127 contain information about the corresponding files in the index. The string item is the file-name, and the numeric items are as follows:

* The file type (a negative value if the file is protected)
  1 = DISPLAY/FIXED data file
  2 = DISPLAY/VARIABLE data file
  3 = INTERNAL/FIXED data file
  4 = INTERNAL/VARIABLE data file
  5 = BASIC program or other "memory image" data
* the total numbers of sectors allocated for the file
* the total number of bytes per record

A type-5 file always has a zero (0) as the number of bytes per record, since this measurement does not relate to memory image data.

What follows is a sample program to illustrate how to display a diskette catalog:

READ THE CATALOG

The following catalog enables you to read and print the catalog for a diskette from TI Basic. Lines 100 through 160 set up a single dimension array of five items corresponding to the five types of files. The next four lines ask for the number of the drive containing the diskette you want to catalog and then check to be sure you have made a valid entry.

```
100 CALL CLEAR
110 DIM TYPE$(5)
120 TYPE$(1)="DIS/FIX"
130 TYPE$(2)="DIS/VAR"
140 TYPE$(3)="INT/FIX"
150 TYPE$(4)="INT/VAR"
160 TYPE$(5)="PROGRAM"
170 INPUT "MASTER DISK(1-3)?":A
180 A=INT(A)
190 IF A<1 THEN 170
200 IF A>1 THEN 170
```

The next section opens the file, reads the diskette information for record #0, and displays it on the screen.

```
210 OPEN#1:"DSK"&STR$(A)&".",INPUT,RELATIVE,INTERNAL
220 INPUT#1:A$,J,J,K
230 DISPLAY "DSK";STR$(A);" - DISKNAME= ";A$:
    "AVAILABLE=" ;K;"USED=";J-K
```

The remainder of the program reads the remaining information in the index, formats it, and displays the catalog on the screen.

```
240 DISPLAY:"FILENAME SIZE TYPE P:
"---------- ----- --------- -";
250 FOR LOOP=1 TO 127
260 INPUT#1:A$,A,J,K
270 IF LEN(A$)=0 THEN 350
280 DISPLAY:A$;TAB(12);J;TAB(17);TYPE$(ABS(A));
290 IF ABS(A)=5 THEN 320
300 B$=""$STR$(K)
310 DISPLAY SEG$(B$,LEN(B$)-2,3);
320 IF A>0 THEN 340
330 DISPLAY TAB(28);"Y";
340 NEXT LOOP
350 CLOSE#1
```

Naturally, you can make some variations to this program. The best place to start is to type it in and get it to run the way it is. Then you may be able to add to it to suit your own requirements. o

continued from page 9

Remember that fuses usually do not blow by themselves - often there is another problem. You can tell if the internal fuse is blown - the PE Box fan works but nothing else does.

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. o

Tips from the Tigercub #63

by Jim Peterson, Tigercub Software, USA

Several articles have been published on the subject of using Funnelweb as a simple fixed-field data base. Sometimes you might want to rearrange the sequence of fields in such a file. This mini-program will quickly change the position of any field in a DV/80 file.

```
100 DISPLAY AT(3,8)ERASE ALL:"FIELDSWITCHER":"":" by
Jim Peterson":"":" To change sequence of fields in a
DV80 fixed field file created by Funlweb or other means"
110 DISPLAY AT(23,6):"PRESS ANY KEY" :: DISPLAY AT(23,6)
:"press any key" :: CALL KEY(0,K,S):: IF S=0 THEN 110 EL
SE CALL CLEAR
120 DISPLAY AT(8,1):"FILENAME? DSK":: ACCEPT AT(8,14):F$
130 OPEN #1:"DSK"&F$,INPUT
140 DISPLAY AT(12,1):"MOVE FIELD STARTING AT WHAT POSITI
ON?" :: ACCEPT AT(13,11)VALIDATE(DIGIT):N
150 DISPLAY AT(15,1):"LENGTH OF FIELD?" :: ACCEPT AT(15,
18)VALIDATE(DIGIT)BEEP:L
160 DISPLAY AT(17,1):"TO WHAT POSITION?" :: ACCEPT AT(17
,19)VALIDATE(DIGIT)BEEP:T
170 IF T>N+L-1 OR T<N THEN 190
180 CALL SOUND(400,110,0,-4,0):: DISPLAY AT(23,1)BEEP:"C
ANNOT MOVE FIELD WITHIN ITS OWN PARAMETERS!" :: GOTO 140
190 DISPLAY AT(19,1):"OUTPUT FILENAME? DSK" :: ACCEPT AT
(19,21)BEEP:OF$
200 OPEN #2:"DSK"&OF$,OUTPUT
210 LINPUT #1:M$ :: M$=M$&RPT$(" ",80-LEN(M$)):: IF T<N
THEN M$=SEG$(M$,1,T-1)&SEG$(M$,N,L)&SEG$(M$,T+1,N-T)&SEG
$(M$,N+L+1,255)
220 IF T>N THEN M$=SEG$(M$,1,N-1)&SEG$(M$,N+L,T-N-L)&SEG
$(M$,N,L)&SEG$(M$,T+1,255)
230 PRINT #2:M$ :: IF EOF(1)<>1 THEN 210 ELSE CLOSE #1 :
: CLOSE #2
240 DISPLAY AT(12,1)ERASE ALL:"ANOTHER? Y/N" :: ACCEPT A
T(12,14)VALIDATE("YN")SIZE(1)BEEP:Q$ :: IF Q$="Y" THEN 1
20 ELSE CALL CLEAR :: STOP
```

And this one will make it easy to completely rearrange the sequence of any number of fields.

```
100 DISPLAY AT(3,9)ERASE ALL:"REARRANGER":"":" by Ji
m Peterson"
110 DISPLAY AT(7,1):"To rearrange the sequence of fields
in a DV80 file of fixed fields created by Funlweb or
otherwise."
120 DISPLAY AT(24,7):"PRESS ANY KEY" :: DISPLAY AT(24,7)
:"press any key" :: CALL KEY(0,K,@):: IF @=0 THEN 120
130 DIM L(20),S(20),F$(20):: CALL CLEAR
140 DISPLAY AT(8,1):"INPUT FILENAME?":"":"DSK" :: ACCEPT
AT(10,4)BEEP:I$ :: OPEN #1:"DSK"&I$,INPUT
150 DISPLAY AT(12,1):"OUTPUT FILENAME?":"":"DSK" :: ACCE
PT AT(14,4)BEEP:O$:OPEN #1:"DSK"&O$,OUTPUT
160 DISPLAY AT(16,1):"HOW MANY FIELDS?" :: ACCEPT AT(16,
18)VALIDATE(DIGIT)SIZE(2):F :: CALL CLEAR
170 FOR J=1 TO F :: DISPLAY AT(12,1):"FIELD #";J;"LENGTH
?" :: ACCEPT AT(12,20)VALIDATE(DIGIT)BEEP:L(J):: NEXT J
:: FOR J=1 TO F
180 DISPLAY AT(12,1):"IN FIELD #";J:"":"PLACE FIELD #" :
: ACCEPT AT(14,15)VALIDATE(DIGIT)BEEP:S(J)
190 IF S(J)<1 OR S(J)>F THEN CALL SOUND(300,110,0,-4,0):
: GOTO 180
200 IF POS(E$,CHR$(S(J)),1)=0 THEN E$=E$&CHR$(S(J)):: GO
TO 220
210 CALL SOUND(300,110,0,-4,0):: DISPLAY AT(16,1):"FIELD
#";S(J);"HAS ALREADY BEEN PLACED!" :: GOTO 180
220 NEXT J
230 LINPUT #1:M$ :: M$=M$&RPT$(" ",80-LEN(M$)):: P=1 ::
FOR J=1 TO F
240 F$(J)=SEG$(M$,P,L(J)):: P=P+L(J):: NEXT J
250 FOR J=1 TO F :: N$=N$&F$(S(J)):: NEXT J :: PRINT #2:
N$ :: N$=""
260 IF EOF(1)<>1 THEN 230 ELSE CLOSE #1 :: CLOSE #2 :: S
TOP
```

If you need to use either of those programs on files with a record length other than 80, just add VARIABLE (or FIXED) and the record length to all the file opening statements, and change that 80 in line 210 or 230.

This subprogram, in which X=28 for a 28-column screen or whatever width you want, will reformat a string of almost any length to print on screen without breaking words, and will return in L the number of lines required to print it, which can be used to space DISPLAY AT statements.

```
31993 SUB FORMAT(X,M$,L):: Y=X
31994 IF LEN(M$)<Y+1 THEN 31996 ELSE IF LEN(M$)<Y+X+1 AN
D SEG$(M$,Y,1)=" " THEN 31996 ELSE IF LEN(M$)<Y+X+1 AND
SEG$(M$,Y+1,1)=" " THEN 31996 ELSE P=Y-1
31995 IF P<1 THEN 31996 ELSE IF SEG$(M$,P,1)=" " THEN M$
=SEG$(M$,1,P)&RPT$(" ",Y-P)&SEG$(M$,P+1,255):: Y=Y+X ::
GOTO 31994 ELSE P=P-1 :: GOTO 31995
31996 L=INT(LEN(M$)/X)-(LEN(M$)/X<>INT(LEN(M$)/X)):: SUB
END
```

The following little program, plus the magic of Funnelweb, should be all the mailing list program that most people would need for home use. Just use Funnelweb to create a file with name on the first line, address on the second line, city and state on the third - or use 4 or even 5 lines for the address if you need to, but the 6th line must either be blank or contain selection codes. These codes can be anything you want, such as C for everyone you want to send a Christmas card to, or B11 to send a birthday card in November, or whatever.

You can put as many codes as you want to on that line, separated or strung together but be sure not to use a code that is part of another code - for instance, if you use B11 for those November birthdays, do not use B or 1 or B1 or 11 for something else.

Then continue with the next address in another block of six lines. Just be sure that the line number of the line just above the first address line is always a multiple of six.

```
100 DISPLAY AT(12,1)ERASE ALL:"Filename? DSK" :: ACCEPT
AT(12,14)BEEP:F$ :: OPEN #1:"DSK"&F$,INPUT :: OPEN #2:"P
IO"
110 DISPLAY AT(14,1):"Print addresses with code -":"":"(
to print all addresses, just press Enter)"
120 ACCEPT AT(15,1)BEEP:X$
130 LINPUT #1:A$ :: LINPUT #1:B$ :: LINPUT #1:C$ :: LINP
UT #1:D$ :: LINPUT #1:E$ :: LINPUT #1:F$
140 IF POS(F$,X$,1)<>0 OR X$ ="" THEN PRINT #2:A$:B$:C$:
D$:"":""
150 IF EOF(1)<>1 THEN 130 ELSE CLOSE #1
```

In Tips #62 I reported on the weird behaviour of the CALL LOAD(-31961,149), when used to clear all defaults and search for a LOAD file on DSK1. I have since found that if you put this CALL at the beginning of a program, it will not execute until an END or STOP is reached - but if you break the program with FCTN 4, it will not be in memory!

I stated that after this CALL LOAD was executed, any number taken to the power of 0 (which should be a value of 1) acquired a value of 220.5727273. I was led astray by the INT in the the formula in which I first found this puzzle. Actually it is 220.57000101, which prints to the screen in the peculiar format F0.57000101.

If a number between 1 and 9 is added to that, it is printed as 1< followed by the number being added, followed by the decimal part. For a number between 10 and 19, the < is changed to = and between 20 and 29 it becomes > (note the ASCII sequence); from 30 to 35 it becomes ? but from 36 to 99 99 the decimal portion is preceded by 0 to 63 respectively. 100 is 2<0.570001 and the pattern continues.

Although these are not valid representations of numbers, they are treated as such. Run a program to give N the power of 2^0, then break the program and experiment in immediate mode.

PRINT N gives that strange F0.57000101. PRINT N+1, or whatever, gives values represented in the format described above. PRINT N*1 will give the true numeric value 220.57000101 but multiplying by some other values gave me results in the odd format, as did dividing.

Peter Walker pointed out to me that trying to subtract from N within a program resulted in printing a value followed by a crash reporting a SYNTAX ERROR (in the line which had just been executed!) followed by a jump to a non-existent line zero!

N-1 should be 219.57.. of course, but in immediate mode PRINT N-1 results in 63.57000101. In the format in which added values are printed, this would be 319.57000101 but the 63.. is actually a decimal value, as can be proved by PRINT CHR$(INT(N-1))! When I tried to get a zero value by PRINT N-64.57000101, the computer blew its mind.

Does anyone know what is going on here?

An IBM program called DOC-SMASH, which sells for about $35, will read a DV/80 file and output it to a printer in full carriage-width lines of elite condensed subscript thereby getting up to 216 lines per page. Bud Wright wrote a TI version, with assembly links, to let us do the same thing for free. His version would not work on my trusty old Gemini 10X, which does not support condensed elite, so I wrote this mini-program which is not as fast as Bud's, but does the job.

```
100 DISPLAY AT(3,9)ERASE ALL:"TEXTSMASHER":"":"For the G
emini 10X printer, to print DV/80 text in lines of 136
characters closely spaced, in subscript."
110 DISPLAY AT(20,1):"Press Enter to end input" :: DIM F
$(20)
120 F=F+1 :: DISPLAY AT(12,1):"FILE #";F:"DSK" :: ACCEPT
AT(13,4)BEEP:F$(F):: IF F$(F)<>"" THEN 120
130 OPEN #2:"PIO",VARIABLE 255 :: PRINT #2:CHR$(27)&CHR$
(83)&CHR$(1);
140 PRINT #2:CHR$(15)&CHR$(27)&CHR$(51)&CHR$(12);:: LN=1
36
150 FOR J=1 TO F-1 :: OPEN #1:"DSK"&F$(J),INPUT
160 LINPUT #1:M$
170 IF LEN(T$)>0 THEN M$=T$&" "&M$ :: T$=""
180 IF LEN(M$)<LN+1 AND POS(M$,CHR$(13),1)<>0 THEN PRINT
#2:M$ :: GOSUB 260 :: M$="" :: GOTO 230
190 IF LEN(M$)=LN THEN PRINT #2:M$ :: GOSUB 260 :: M$=""
:: GOTO 230
200 IF LEN(M$)<LN AND EOF(1)<>1 THEN LINPUT #1:X$ :: M$=
M$&" "&X$ :: GOTO 170 ELSE IF LEN(M$)<136 THEN PRINT #2:
M$ :: GOSUB 260 :: GOTO 240
210 P=LN
220 IF SEG$(M$,P,1)=" " THEN T$=SEG$(M$,P+1,255):: M$=SE
G$(M$,1,P):: PRINT #2:M$ :: GOSUB 260 :: M$="" :: GOTO 2
30 ELSE P=P-1 :: GOTO 220
230 IF LEN(T$)<LN+1 AND POS(T$,CHR$(13),1)<>0 THEN PRINT
#2:T$ :: GOSUB 260 :: T$=""
240 IF EOF(1)<>1 THEN 160
250 CLOSE #1 :: NEXT J :: STOP
260 X=X+1 :: IF X<121 THEN RETURN
270 X=0 :: FOR K=1 TO 8 :: PRINT #2 :: NEXT K :: RETURN
```

For that to work properly, your paragraphs must end in carriage returns, and so must the title line, etc. If such is not the case, try Bill Wood's method - load the file into Funnelweb, enter RS for Replace String, then /. /.X/ but instead of X type CTRL U SHIFT M. At the first prompt, enter A for All. If your text has any paragraphs ending in ? or !, get your cursor back to the beginning, change that first period to ? or !, and do it again. You might also need to manually add carriage returns to titles, etc. Just type CTRL U, then SHIFT M wherever a CR is needed.

Without having printers to test it on, I think the program can be modified for the SG-10 by changing line 140 to

```
140 PRINT #2:CHR$(27)&"B"&CHR$(4)&CHR$(27)&CHR$(51)&CHR$
(12);:: LN=160
```

And for old Epson-type printers which do not support elite condensed by

```
140 PRINT #2:CHR$(27)&CHR$(77)&CHR$(27)&CHR$(51)&CHR$(18
);:: LN=132
```

And new Epson compatibles by

```
140 PRINT #2:CHR$(27)&CHR$(77)&CHR$(15)&CHR$(27)&CHR$(51
)&CHR$(18);:: LN=160
```

You might also have to change that 8 to 12 in line 270 - my old Gemini seems to think that 11*12=128.

COMPLETELY out of memory,
Jim Peterson

Tigercub Printall version 1.6

by Jim Peterson, Tigercub Software, USA

This program will print your text in a choice of 1 to 5 columns, and gives you complete choice of fonts, left and right margins, spacing between columns, lines per page, etc, etc. I think the prompts are self explanatory.

NOTE: Some folks have thought that this program did not work because they expected it to reformat text into the desired column width. Use Reformatter+ or the Funnelweb Formatter to do that.

It takes some time to read in text and format it into multiple columns, so if you need to print more than two copies, or will need more copies in future, it will pay you to print back to the disk. To do this, at the printer prompt type over the PIO.LF default with DSK. and a drive number and file name. The text will then be formatted and printed to a D/V 254 file.

The next prompt is for the record length, which will be the default of 80 if the text was prepared with TI-Writer or whatever. However, if you enter 254 you will be prompted for an input file name of a file printed to disk by this program, and for the number of copies wanted, which will then be printed immediately.

If you have Triton's Super Extended BASIC module, you can LIST an Extended BASIC program to disk in 28-column format by LIST "DSK1.filename":28:1-32766. The result will be a D/V 28 file. With this program you can print the listing in 5 columns by selecting 28 record length, elite condensed, 5 columns, 28 column width.

This version has been modified slightly so that it will allow the use of "Control U" codes input by Funnelweb, to underline, emphasize, double strike, etc. an individual word or line. Note that if you are printing in more than one column you must turn off the codes at the end of the line, or they will also affect the same line in all subsequent columns. You must also remember that the control codes will be deleted in printing, which will affect the format.

If the file has a Tab setting, first enter T to get to the tab line, place a period to replace the R, then go to the end of the tab line and place an R, so that lines can be shoved right. For this reason, CTRL U codes cannot be used with full 80-column lines.

For instance, if you want to underline a word, press CTRL 0 to get the open cursor fixed mode. Position the cursor on the first letter of the word, type FCTN 2 and then space bar 3 times to open up 3 spaces, backspace to the first of these, and type CTRL U, FCTN R, CTRL U, -, CTRL U, CTRL A, CTRL U. Move the cursor to the first space beyond the word, type FCTN 2, space 3 times, backspace 3, type CTRL U, FCTN R, CTRL U, -, CTRL U, CTRL , CTRL U.

If the word is at the end of a line or you are underlining a complete line, and the line is not completely filled with characters, go to the end of the line first and put the "turn-off" codes starting in the space just after where the last character would be. For instance, if the column width is 40, start the codes in column 41.

continued on page 23

XB tips Number 19

by Jim Swedlow, CA USA

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

ON FREEWARE

Last year I wrote an Extended Basic program that prints a Multiplan spreadsheet sideways. I wrote it because I was tired of seeing new products for those other computers but not the 99/4A.

After a great deal of thought, I decided to release it as freeware. I have shipped over 100 copies of my program and, based on user input, revised it twice.

The experience exceeded my expectations. I have had requests from over 30 states and Canada and have received some very nice letters. Its nice to know that there are others out there who still use the 4A.

I would like to pass on some suggestions should you write for freeware.

IF YOU SEND MONEY: First and foremost, please print clearly. I agonized over deciphering two addresses. Better yet, if you have one, send a printed return address label. You could use this month's program to make some.

IF YOU SEND A DISK: Initialize the disk. This makes sure that it is OK. Sweeping the disk is not sufficient, format it and verify the sectors.

Then check with the Post Office before mailing. Most mailers weigh under 2 ounces which is $0.39. I have received mailers with up to and over one dollar for postage!

While you are at it, why not put a few public domain programs on the disk? Three folks sent me some programs that were really nice. Do include a note so that the recipient will check the disk, however.

WRITE A NOTE: I released my program to share something with the TI community. The notes I received were worth their weight in gold. On the other hand, the guy who sent me a check without any kind of note did not make any points!

GIVE FEEDBACK: Take the time to let the programmer know what you thought of the program. If you liked it, say so. If it did not work, let him know. If you have a suggestion, pass it on. If you want a response, include a SASE (Self Addressed Stamped Envelope). The whole thing costs you $0.44 postage but is important to the recipient. Believe me!

PAY FOR THE PROGRAM: Some freeware/shareware programmers ask you to pay $10 or so if you like the program. Considering that most commercial software starts at $20, this is a bargain. Some of the best new items come from freeware/shareware. Without your support, this source just might dry up.

PS: This also applies to freeware/shareware from our users group library. If the program or documentation asks for money, send it. The $2 charge per disk goes to our users group, not the author.

ON PRE-SCAN AND DIM

If you are not familiar with pre-scan, read the supplement to the Extended Basic manual and my column in the June, 1985 ROM.

Your TI does not execute a DIM statement. The only time it reads the DIM is during pre-scan. Therefore, DIM can be part of the pre-scan list.

For those arrays that have ten values, you do not need to DIMention them. You only need to put A() in the pre-scan list. The parenthesis can be completely empty (assuming a one dimention array, that is).

This, then, is a legal, functional pre-scan list:

```
100 GOTO 110 :: A,B,C$,C(),D$ :: DIM Q(12) :: !@P-
110 ! Program continues
```

This program, of course, uses no subroutines. For another example, see lines 150 to 180 of this months program.

99-CALC

I just received a copy of 99-CALC, a freeware spreadsheet by Phil Barnes. I will let you know how it works as soon as I test it out.

Much to my surprise, I discovered that Phil authored the original disk label program that came from our user group library and was the basis of this month's program and one I published here previously. I did not note the source before because I did not know it. This should correct that error.

Should you want 99-CALC right away send an initialized disk, a self-addressed mailer and return postage to:

Phil Barnes
24631 Via San Fernando
Mission Viejo, CA 92692

LABEL

I have not published a program in a few months -- my entry into the freeware arena took up more time than I expected.

I have previously published this program as two separate ones: one that prints disk labels and one that did mailing labels. It got to be a chore changing programs so I combined them.

CUSTOMIZING: Substitute your name and address in the DATA statements in line 170. Be sure and keep the leading and trailing commas (unless you use a four or five line address). Your address lines should be no more than 27 characters long.

Lines 190 to 260 are the printer control codes. They work for most Epson and compatible printers. If you printer uses different codes, change them as necessary.

If your printer name is not PIO, modify line 280 as needed.

DEBUGGING: If you run into problems, delete the <:: !@P-> at the end of line 180 to keep pre-scan on for the entire program and change <ON ERROR 580> to <ON ERROR STOP> in line 370. This should help you find any errors.

PROGRAM NOTES: Lines 550 and 560 do some interesting things:

ON POS is used to validate the correct key return. The value of I then is used, after returning from the subroutine, as the value in <ON I GOTO . . .>.

The second CALL KEY in line 560 remaps the keyboard as a 99/4A so that you can input lower case for your labels.

By changing 13 and 32 to 80 (or <ENTER> and <SPACE BAR> to <P>) in line 550, you can start printing by pushing P or ENTER or the SPACE BAR.

Enjoy.

continued on page 20

Harnessing the Power of Speech

by Craig Dunn, England

The TI Speech Synthesizer is an amazing little device. It was a breakthrough for the lower end (priced) computers. Unfortunately, many 99/4A owners still do not know how to access speech along with its little features. Sure, a lot of games use speech to add interest and excitement, but the applications of speech goes far beyond games.

One of the major features of the speech synthesizer is its ability to let you ad speech to your programs. There are several ways to do this, including TI's Terminal Emulator II (hereinafter called TEII), Extended Basic, and through the use of Assembly Language routines. Extended Basic provides a rather limited vocabularly, unless you are using one of the recent utilities that give you unlimited speech in Extended Basic, but that is another story. TEII allows for unlimited speech directly from Basic. This built-in text-to-speech capability of TEII will be the focus of this article.

First, plug in the TEII command module, turn on the computer, and select TI Basic. Now type and run the following program:

```
100 OPEN#1:"SPEECH, OUTPUT"
110 INPUT A$
120 PRINT #1:A$
130 GOTO 110
```

If you get an error, make sure you have the Speech Synthesizer connected properly to the side port. Now we have a very simple text-to-speech editor. Line 100 contains the OPEN command needed to access TEII speech capabilities. Line 120 sends the text strings that you type in to the text-to-speech interpreter, which then sends the information to the synthesizer. Experiment with this awhile by typing in phrases, followed by an <ENTER>.

In the above example you were in the default speech mode. This means that no commands have been sent to alter the voice. We can change the voice easily using the "//" command. The proper format is:

// pitch slope

e.g. //34 118

The pitch is a number between 0 and 63. A zero causes the speech synthesizer to whisper phrases. Pitches from 1 to 63 range from the highest pitched [1] to the lowest pitch [63]. For the best sound, figure the SLOPE using the following formula:

SLOPE = 32 X (PITCH/10)

Round this result to the nearest whole number. Now, when you enter the command along with these two numbers, it will appear that nothing has happened. But type in a simple phrase and press <ENTER>. You will notice the change in voice. For example, st the prompt in our simple little speech editor, type "//55 176" and press <ENTER>. Be sure to include a space between the numbers. Nothing happened, right? Well, now type something in and press <ENTER>. See how the voice changed? It becomes deeper. Now try "//0 0" and press <ENTER>. Again type in a short phrase. Another voice tone! Experiment with these and other PITCH/SLOPE combinations to get the feel of working with these.

Before we wrap up this tutorial, we will take a look at the INFLECTION symbols. The symbols are " " (caret), "_" (underline), and ">" (greater than). The " " when placed in front of a word, indicates a primary stress point to the test-to-speech interpreter. Only one " " may be sued per string. The "_" is used to indicate a secondary stress point and may be used without limit throughout the string. The ">" will shift the stress points within the word. Experiment with all of these to make words sound better and more human like. Remember, all inflection symbols must precede the word they are to affect.

One final note- remember that the test-to-speech interpreter is not perfect. Sometimes you might have to alter a word's spelling drastically to make it sound right.

o

Newsletter Update

by Bob Relyea

TIBUG (Brisbane), May, 1992: Basic Computers; Bits and Pieces; Book and Software Library; Debugging; DSK.Diskname.Filename; Editorial; GUI? Phooey!; In the P.O. Box; Meeting News; Letter to the Editor; PagePro 99 & Utilities; Shop; Tips #39; Trading Post; What's News?; Word Processing #2; You Do Not Have To Have It All.

OTTAWA, May, 1992: President's Message; Appreciate Your Programmers; Why Should You Learn To Program?; Programming Music The Easy Way, Part 2; Fast Extended Basic.

Spirit Of 99, May, 1992: Meeting Minutes; BBS Tax Rumour;Don't Let Your TI-99/4A Become A Pain In The Neck; More Pain In The Neck; Gen-Tri; Pix of Pixels; Easy Sort, Assembly Language of XB Data Statements; TI still cares; Lima Multi User Group Conference; My Geneology Program; MDOS Buyout; Tom's Observations; About The D.O.M.; Assembly Language Lesson 2; Letter Writing; Harnessing The Power Of Speech (Part 1).

UGOC ROM, May, 1992: Editor's Article; President's Message; April Board Minutes; The Member Ship; Slate of Nominees; Dips, Chips and Sips; Dark Is Not Dead; Compubits (again); XB Miscelany #13; Our TI*Two; For/Next Loops; Painting A Wall; UGOC Newsletter Exchange List.

June, 1992: President's Notes; May Board Minutes; Editor's Message; The Member Ship; Blank Wall; Clearing House BBS; Chaos, Fractals, etc; Hole Currents; XB Miscelany #14; Graphical Comment; A Friend In Need.

THE FRONT RANGER, October, 1991: President's Corner; Meeting Minutes; Breaking The 40 Column Barrier; Routine Interweaving; Tiny Lotto; A System Search Program (XB).

November, 1992: President's Corner; I Like Brain Games; Why Should You Learn To Program?; Computer Purchase Fraught With Peril, Dismay; Club Program Listings.

December, 1991: President's Corner; You Don't Have To Have It All; Meeting Minutes; Fixing a TI-RS232 ; TI Fest West, 1992; Small Business Report; Club Program Listings.

THE PUG PERIPHERAL, February, 1992: Club News; Gen-Tri; From The Librarian; Writing Articles; You Can't Do This - Oh Yes You Can!; Tips #61; Tournament Solitaire.

March, 1992: Club News; From the PUG BBS; From the Librarian; Disk Of Horrors; Meeting Minutes; TI Still Cares; Gen - Tri, a Review; I Like Brain Games.

THE BOSTON COMPUTER SOCIETY, April, 1992: Listen; TI Casino; TI - 101; #1 Corpus; Software Program Reviews.

BYTEMONGER, May, 1992: Standards; Meeting Notes; From the Soap Box; In My Own Words; Gram User Menu System; Seahorse Software Update; CRU Addresses; Lima Update.

o

continued from page 8

```
320 FOR S=1 TO F :: FOR J=1 TO F
330 Z=Z+1 :: K=S+J-1
340 P(S,J)=K-F*INT(K/F):: IF P(S,J)=0 THEN P(S,J)=F
350 Q(Z)=P(S,J):: NEXT J :: NEXT S
360 FOR X=1 TO F :: Y=F*X-F+1 :: FOR C=1 TO F-1
370 Y=Y+1 :: DP(X)=DP(X)*D(Q(Y)):: NEXT C
```

o

To See or Not to C

by Geoff Trott

I hope you had success with the program from last month. It is important that you try things out if you are to learn a new language. As I said at the beginning (I think I did), there are two problems to be overcome, learning the language and learning how to use the editor, compiler, assembler and loader. The first of these is hard enough so that is why Funnelweb is such a good idea to use to make the second so much easier than doing all those steps by hand (as it were). To use Funnelweb to best effect, you must set up a script loader file which lists the programs to be loaded and which can also be used to assemble all those files also. Anyway, now we should look at the rest of the procedures in the string "library". Here are the brief descriptions of these functions in STRINGDOC.

Find the position of a character in a string

```
index(s, c)
char *s, c;
```

This returns an integer of the position of the first occurrence of c in s. If c is not present in s then zero is returned.

Find the right most postion of a character in a string

```
rindex(s, c)
char *s, c;
```

This returns an integer of the position of the first occurrence of c in s, starting from the right end of s. If c is not present in s then zero is returned.

Concatenate two strings

```
strcat(s1, s2)
char *s1, *s2;
```

Join s2 onto the end of s1. The new string is returned in s1.

Concatinate n characters of s2 onto s1:

```
stncat(s1, s2, n)
char *s1, *s2;
int n;
```

At most n characters of s2 are joined to the end of s1. The new string is returned in s1.

Copy one string to another

```
strcpy(s1, s2)
char *s1, *s2;
```

Copy s2 into s1. The new string is returned in s1.

Copy n characters of s2 into s1

```
stncpy(s1, s2, n)
char *s1, *s2;
int n;
```

At most n characters of s2 are copied into s1. The new string is returned in s1.

These procedures provide a useful set for doing a range of functions on strings or arrays of characters. There are the two functions for finding the position of a character in a string, looking from the beginning and from the end. These two procedures take as inputs the pointer to a string and a character and return an integer as a result. The next two procedures deal with merging two strings together. The first merges the two strings into one, putting them both into the first string. The second of these procedures merges a number of characters of the second string onto the end of the first string, with the resulting string replacing the first string. These two procedures have two pointers to strings as inputs, with the second one also having an integer as well, with no result returned as the first string is replaced with the result. The last two procedures copy (or move) the second string into the first string. The first procedure moves the whole string across, while the second one only moves a given number of characters across. The inputs and outputs are the same as the previous two functions.

I am sure that you will think of many other useful procedures for strings (equivalents for RPT$ and SEG$ spring to mind) which you could code and add to the library. That gives me an idea for an interesting exercise. We will see how we could write the C eqivalents for these two functions. But first we will look at the above functions and see how they do their tasks.

```
index(s, c) /* returns location of c in s */
char *s, c;
{
  int n;
  for (n = 1; *s != NULL; n++, s++) {
    if (*s == c) return (n);
  }
  return (0);
}
```

For this procedure there are two input parameters; a pointer to the string and a character. These are declared first and then the procedure entered. A local integer variable is declared to do the counting and this is done in a for loop. The for loop initialises n to be one and increments both n and s (the pointer to the characters in the string until the NULL character is reached (the end of the string character). The test here is the character pointed to by s is not equal (!=) to c. Each time through the loop a match is looked for between the string character and the input character. If a match is found, the subroutine ends and the value of n is returned. If the for loop ends without the match occurring, zero is returned. This is a simple function.

```
rindex(s, c) /* returns location of c in s from right */
char *s, c;
{
  int n;
  for (n = 0; *s != NULL; n++, s++) ;
  s--;                /* point to last character */
  for ( ; n > 0; s--, n--) {
    if (*s == c) return (n);
  }
  return (0);
}
```

This does the same thing as the previous procedure but from the other end of the string. To do this the pointer is first moved to the end of the string with the value of n being incremented. Then another for loop is used with the variables n and s decremented while the match is checked. If a match is found the value of n is returned or if no match is found, n goes to zero and zero is returned by the procedure.

```
strcat(s1, s2) /* concatenate s2 to end of s1 */
char *s1, *s2;
{
  while (*s1++) ;
  s1--;
  while ((*s1++) = (*s2++)) ;
  return;
}
```

This procedure concatinates two strings together, that is it puts one string onto the end of the other one. In this case, the second string is put on the end of the first one. In this procedure, there is no need for a local variable. The while loop is used to move the pointer s1 to the end of the first string. This uses the fact that FALSE is the value of zero while TRUE is any non zero value. In C, putting the name of a variable gives the value of that variable. The instruction which is executed while the while test is

TRUE is a blank instruction (that semi-colon) and the *s1++ does all the work of incrementing the pointer to point to the next character in the string, and getting the actual character, which will be non-zero until the end of string character is reached. s1 has been post incremented, that is the pointer is incremented after the character has been fetched. This means it will point to the character after the end of string character, so it is decremented before entering the next while loop. This loop then puts the string pointer to by s2 on the end of s1 and the value is non-zero (TRUE) until the end of string character of s2 is reached. The string is returned in the parameters so there is nothing to return at the end.

```
stncat(s1, s2, n) /* concatenate at most n chars of s2
                                        to end of s1 */
char *s1, *s2;
int n;
{
  while (*s1++)
    ;    /*point to end of s1*/
  s1--;
  for ( ; n > 0; n--, s1++, s2++) {
    *s1 = *s2;       /* move s2 to end of s1 */
    if (*s1 == NULL) return;
  }
  *s1 = NULL;
  return;
}
```

This is a very similar routine to the previous one, except that at most n characters of s2 are appended to the end of s1. The first part of the routine is the same and it is only the second while loop which has become a for loop to count the characters which has changed. This uses the value of n as a counter and n is decremented along with s1 and s2 being incremented each time through the loop. This shows some of the power of C in that the three variables n, s1 and s2 are all changed automatically by including them in the correct place in the for statement. Also you can initialize a number of different variables at the start of the loop (or none as in this case).

```
strcpy(s1, s2) /* copy s2 into s1 */
char *s1, *s2;
{
  while ((*s1++) = (*s2++)) ;
  return;
}
```

Copying one string into another is a simple function which uses a while like strcat(). You should be able to understand this if you have understood the previous two.

```
stncpy(s1, s2, n) /* copy at most n characters
                                  of s2 into s1 */
char *s1, *s2;
int n;
{
  for ( ; n > 0; n--, s1++, s2++) {
    *s1 = *s2;
    if (*s1 == NULL) return;
  }
  *s1 = NULL;
  return;
}
```

To copy at most n characters of one string to another is a bit trickier. I have made it a bit longer than it needs to be but it makes it a bit more readable in the process. A for loop is used to make sure that at most n characters are copied. Each time through the loop the presence of the end of string character is checked and if there the exit is taken. If the loop finishes without finding an end of string character, one is appended to the end of the string to make it a valid string.

This concludes the string procedures that are in the release of c99 that you can get through the shop. If you have another version you might like to compare the two versions. Better yet, you should try them out with a test program and see if they have any problems. That is what I did and this led me to re-write some of them to get correct operation for all inputs. Now let us see if we can write some procedures to emulate the BASIC routines of RPT$ and SEG$. The other string routines are already available in c99. For example, ASC and CHR$ are already available as C can treat characters as numbers or characters and the ASCII code is the one that is used to translate from one meaning to the other. LEN is one of the procedures already mentioned. POS is partially done with index(). This could be expanded to cater for the position of a string rather than just a character. STR$ and VAL are done on input and output with the format characters and scanf() and printf(). They could be done with procedures to do the conversion between a character string and a number but c99 does not have floating point built in so they are not particularly difficlt for integers. Standard C has a function atoi() which converts an ASCII string to an integer. I do not think this function is in the c99 library. Here is another one to write along with the RPT$ and SEG$ procedures.

Let us first look at RPT$. This is a procedure to produce a string of n characters all the same. In BASIC we would have something like A$=RPT$("ABCD",4) and this would give the variable A$ the value of "ABCDABCDABCDABCD". In C we cannot return a string as a value from a function so we would need a procedure call like repeat(s1, s2, n) where s1 would be a pointer to the output string, s2 would be a pointer to the string to be repeated and n would be an integer which would be the repeat value. The procedure would simply copy s2 to s1 once and then concatinate s2 onto the end of s1 n-1 times. This would be done in a for loop which would make use of two procedures that we already have, along with some checking of values to avoid errors. See if you think this would work.

```
repeat(s1, s2, n)
char *s1, *s2;
int n;
{
  int max, i;
  if (n > 0) {
    strcpy(s1, s2);
    max = n * strlen(s2);
    if (max > 255) max = 255;
    max = max / strlen(s2);
    if (n > 1) {
      for (i = 1; i < max; i++)
        strcat(s1, s2);
    }
  }
  else
    *s1 = NULL;
  }
  return;
}
```

There is a bit of fussing to make sure that the resulting string is less than 255 characters, which it must be in BASIC but does not have to be in C. This is because in BASIC the first byte in a string is a byte containing the length of the string. Eight bits can only hold a number as large as 255. In C the string is determined by the end of string character and so this limit does not apply. It is just as well to have some limit for a procedure like this and 255 seems a good number. Note that the calling procedure must supply a character array of the right size to hold the string. c99 does not check that it is using space not allocated to that array! (Most C compilers do not check for these sorts of errors.) The user must be aware of the potential problems. Another potential problem is if you want to repeat a single character. You must make sure that the character is a single character string (with the end of string character) rather than just a single character with no end of string character. This problem arises because C has a data type of char, or single character, while BASIC does not. It only has the string data type (name ending with $) when dealing with characters.

continued on page 22

TI-Base Tutorial #18

by Martin Smoley, USA


```
     * ---------------------------------- *

*                        RENEW/C
CLOSE ALL
SET TALK OFF
CLEAR
SET RECNUM OFF
SET HEADING OFF
SET PAGE=000
LOCAL TEMP C 40
LOCAL COD C 5
WRITE 8,4,"Prints Labels to RS232 and"
WRITE 10,4,"Prints RENEW letter to PIO."
WRITE 12,4,"Set Printers + press ENTER"
WRITE 19,4,"Print dates prior to: YY/MM"
WRITE 21,4,"Example date is: 90/01"
WRITE 23,4,"Enter your date:"
READSTRING 23,21,COD
CLEAR
  WRITE 10,10,"Opening DSK1.TNAMES "
   SELECT 2
   USE DSK1.RENEW
  WRITE 10,10,"Opening DSK1.RENEW  "
   SELECT 1
   USE DSK1.TNAMES
  WRITE 10,10,"**** Searching **** "
   WHILE .NOT. (EOF)
    IF (XP<COD).AND.(XP<>" XPDt");
     DO DSK1.RENLBL
     DO DSK1.RENLET
    ENDIF
    SELECT 1
   MOVE
   ENDWHILE
CLEAR
SET TALK ON
CLOSE ALL
SET RECNUM ON
SET HEADING ON
RETURN Copyright Martin A. Smoley 1990
*
* Main CF to print Mailing Label and
* Custom Letter to remind members that
* their membership needs renewing.
*
* USEs DBs TNAMES and RENEW
```

```
   CREATED 02/13/90 CHANGED 03/11/90
  FIELD  DESCRIPTOR  TYPE  WIDTH  DEC

     1   NM           N     004    00
     2   LINE         C     076

        000 1 RENEW   00009/00011eof
```

```
REC  NM  LINE

0000  10    We at the NorthCoast 99'ers User Group are sorry to see
0001  20  that your membership in our group is about to expire. We
0002  30  hope that this is an oversight on your part and that you
0003  40  merely forgot to send in your fifteen dollars. As a group
0004  50  we expect to have another great year and hoped that you
0005  60  would remain a member and join in the activities. Please
0006  70  stay with us and send in your fifteen dollar membership
0007  80  renewal as soon as possible.
0010  85
0008  90                                  Your friends,
0009 100                                  The NorthCoast 99'ers.
```

This month I am going to show you how TI-Base can produce moderately personalised letters and mailing labels with little or no effort. It is the reverse version of a previous tutorial. In that tutorial I used TIB to set up a Mailmerge file that could be used by FunnelWeb to send personalised letters. The minor drawback in that case is that FunnelWeb does not have any decision capabilities. It sends the letter to everyone, or to those you choose. Also, it does not do mailing labels as easily as a database program. In contrast to that, TI-Base will produce a letter and mailing labels simultaneously, provided you have two printers hooked up at the same time, and the letter can be moderately personalised depending on the amount of work you are willing to invest.

The first thing I did was write a demo letter, which you see above. I wrote the letter and edited it in FunnelWeb. I included the Formatter commands .LM 5;RM 62;FI;AD at the top of the page. Then when I formatted the letter, I entered DSK1.RENEW in place of PIO. This sent the file to disk. When the Formatting was finished I edited RENEW and deleted all of the LF and CR symbols along with removing all the extra blank lines. I also added the column of numbers, 10 to 100, in the blank space to the left of the lines. After that I used PF, for Print File, to print the file back to DSK1.RENEW. TI-Base will CONVERT this file into a Database very nicely.

At that point I fired up TI-Base and entered CONVERT DSK1.RENEW GO. When TIB gave me the CREATE screen I entered NM, N, 4 and 0 for the first field. Then I entered LINE, C, and 76 for the second field. The output width of the Formatter is 80 columns, so the total field widths should be 80. At that point I pressed <FCTN 8> and TIB created a DB named RENEW. At this time RENEW is not usable. You must enter USE RENEW and, after it seems to have loaded, you must enter RECOVER. At that point TIB will finish the job and you will have a completed database named RENEW. I sorted RENEW on the NM field. This allowed me to add blank lines or possibly some text by appending a line and then numbering it between two existing numbers. An example of this is the blank line number 85, which is actually record number 10.

The letter DBs will be easy to do after you have tried a few. Once you get the hang of CONVERT and RECOVER, these too will seem like child's play. The basic CFs that you need to make the system work are RENEW, RENLBL and RENLET, which I have included. RENEW asks you to enter the date prior to which letters will be sent. It is in the form 90/01, or year/month, which matches the field in the DB TNAMES. You can use another style or you can search for specific dates as we did in a previous tutorial. As I have said in the past, start with a couple small CFs and build on them until you have a big system that does exactly what you need.

As RENEW searches the DB it will run (or DO) RENLBL and RENLET every time it finds a record with an expiration date. RENLBL switches the printer to RS232.CR.LF.DA=8, my serial port TI-Impact printer for labels, and prints out one label. Then it switches back to the PIO.CR.LF printer, which is my Star NX-10 for the letter. RENLET prints the letterhead, the current date, the persons name and address, "Dear ?????", and the letter. NOTE: I have printed a condensed, finished letter at the bottom of this column for your inspection. I have not presented a label because of a lack of space.

The CFs and DBs in this tutorial are relatively straightforward. There is some confusion in using a page length, so I set the page length to 000, which means no page eject, and I issued a FF, Form Feed, at the end of each letter. Remember, I am using two printers and any page length would mess up my labels. If you only have one printer, make two copies of RENEW, (RENEW1 and RENEW2). In RENEW1, only DO DSK1.RENLBL and in RENEW2 DO DSK1.RENLET. This will allow you to run labels and then convert to paper and run the letters. If you enter the same date, you will get the same names.

This system could be enhanced to send out little notes to sales contacts, if that is what you need. You could say "Hi! How are you doing. I have not heard from you in a while. If you had a field in the database for that person's children or birthday, TIB could test for those facts. If found, TIB could add "PS: By the way, how are the kids" or "By the way, Happy Birthday". This type of stuff could really make the letter seem like a personal contact. If you have the imagination, your TI can do the job.

NorthCoast 99'ers
6149 Bryson Drive
Mentor, Ohio 44060

03/12/90
Grant E. Aardvark
9995 State Rt. 84
Geneva OH. 44014

Dear Grant

We at the NorthCoast 99'ers User Group are sorry to see that your membership in our group is about to expire. We hope that this is an oversight on your part and that you merely forgot to send in your fifteen dollars. As a group we expect to have another great year and hoped that you would remain a member and join in the activities. Please stay with us and send in your fifteen dollar membership renewal as soon as possible.

Your friends,
The NorthCoast 99'ers

```
*                          RENLBL/C
LOCAL TEMP C 40
SET PRINTER=RS232.CR.LF.DA=8
  WRITE 10,10,"** Printing label **"
IF .NOT. (EOF)
   REPLACE TEMP WITH "          ";
     | "       Exp. Date " | XP
     PRINT (E),TEMP
     PRINT (LF)
   REPLACE TEMP WITH TRIM(FN) | " ";
           | MI | " " | LN
     PRINT TEMP
     PRINT SA
   REPLACE TEMP WITH TRIM(CT) | " ";
           | ST | ". " | ZP
     PRINT TEMP
     PRINT (LF)
 ENDIF
SET PRINTER=PIO.CR.LF
RETURN Copyright Martin A. Smoley 1990
*
* Prints one Mailing Label for RENEW.
*
* SETs PRINTER to RS232.CR.LF.BA=8 and
* back to PIO.CR.LF
```

```
*                          RENLET/C
LOCAL HTEMP C 60
  WRITE 10,10,"* Printing letter * "
IF .NOT. (EOF)
   REPLACE HTEMP WITH " ";
     | "     NorthCoast 99'ers"
     PRINT (E),(e),HTEMP
   REPLACE HTEMP WITH "          ";
 | "           6149 Bryson Drive"
     PRINT HTEMP
   REPLACE HTEMP WITH "          ";
 | "           Mentor, Ohio 44060"
   REPLACE TEMP WITH .DATE.
     PRINT HTEMP,(LF),TEMP,(LF),(LF)
   REPLACE TEMP WITH TRIM(FN) | " ";
           | MI | " " | LN
     PRINT TEMP
     PRINT SA
   REPLACE TEMP WITH TRIM(CT) | " ";
           | ST | ". " | ZP
     PRINT TEMP
     PRINT (LF),(LF)
ENDIF
```

continued on page 4

Making a TIPS Label Letterform

by Earl Raguse, UGOC, CA USA

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

I have used TIPSLABEL to make border letterforms several times, but I always manage to spoil a few while re-learning what to do, so I am writing it down. Maybe you too can profit from it.

Firstly, it is all done with the Image (only) Option, except the last part, the actual letterhead, which is done with the Text (only) Option. It is assumed you know how to work TIPSLABEL, this is not a beginning tutorial.

1. First, be sure your printer and paper are at TOF (Top of Form), and then (re)-positioned to the line where you want the images to be printed. On all my printers, the way to be sure that that position is in the printers memory, is to switch power Off/On. I will use TOF to refer to this position in the following, even when it is for printing at the bottom.

2. Select the image for the top row.

3. Select Options, then Multiple.

4. Change the default column 5 to 9.

5. Accept the prompt of 1 space between images.

6. Enter 7 images, no more, no less.

7. Now 7 images should be printed across the top of your paper, starting at columns 9 and ending at column 65.

8. Now select FF from the menu to move you to the TOF of next page.

9. Then select Again, from the menu, then 1 to print another.

10. I like to at least a dozen sheets at a time, so I repeat steps 8 and 9 but until you are proficient maybe you had better stick to 1 or 2 pages.

11. Rewind your printer to TOF, then move paper up until you are about ten lines above the bottom. Switch power Off/On again, very important, this is the new TOF.

12. If you plan to put a different image at the bottom of the page, do FCTN 4 (BREAK) and RUN. This is to prevent TIPSLABEL from printing an underline under all EVEN numbered images. I will tell you an easier way around this in an article on TIPSLABEL, if I ever get it written, else select Again 1, and go back to 8, else continue.

13. If you now have multiple images on top and bottom of your letterform, let's proceed to do the sides.

14. Again do BREAK RUN, select your new image, then Options, then Image (only), then Single.

15. You will then be asked At What Column to Put it?, enter 00.

16. Next answer 1, to the How Many query. This is a test. The image will be printed on the left side of the paper. If you got your paper lined up right, it will be in linewith the previous images, if not, adjust. Then Select Put from the menu, 00, then 10, and you should get images all down the left side.

17. If you need to print more, select FF, do Put 00,11, and repeat until you have done enough.

18. Rewind the paper to TOF. Now repeat steps 14-17, except PUT the image at 71.

continued on page 2

Computaholics Exam

by Robert Brown

Are you a Compute-a-holic? Many of us are without even realizing it. Below are a FEW questions to help you take a somewhat humorous look at your computing habits, and decide if you need help.

Questions

1. Do you use disk labels for tapes?

2. Do you buy disks in lots of 100?

3. Has this ever happened to you?... Your spouse gets frisky and you say, "Not tonight honey, I have got a bug." And then you have to explain that you meant in your program...

4. Do you use more than a box of computer paper per year?

5. Do you own and frequently use a calculator capable of Hexadecimal and binary arithmetic?

6. Do you have computer style personalized license plates such as: CPU-HED? PLA-PHA? PAG-ZRO? MEG-4ME? GIG-BYT? TI994A? I1T-RAM?

7. Can you look at memory hex dumps and disassemble them in your head?

8. Have you ever written an assembly language program that is more than 10K of pure object code?

9. Every time you pass a computer, typewriter, or anything with keys, do you get this irresistible urge to type something?

10. Is your profession non-clerical in nature, yet you can type 70 words per minute or more?

11. Would you RATHER write a video game than play one?

12. When a friend calls you to ask you about a problem with his monitor, do you immediately start thinking in assembly code, when he meant his Monochrome display?

13. Do you HOPE the teacher assigns an essay instead of an exam, so you can do it on your computer?

14. Have you ever waked up at 3am, face down on your computer's keyboard?

15. Do you take computer magazines to the toilet with you?

16. Do you have a bumper sticker which reads, "I would rather be COMPUTING than COMMUTING"?

17. If you are a professional programmer do you ever wonder, "Why do they PAY me to have this much fun?"

18. When lunch rolls around, do you start working on a computer program for home?

19. Do you have a "Computer Room" in your house?

20. Do you take computer books to the doctor's office, dentist, or barber with you?

21. Do your kids teethe on game cartridges and disk boxes?

22. Do your kids use your bad disks for frisbees?

23. Do you subscribe to more than one COMPUTER MAGAZINE?

24. Do you ever take a holiday, so you can spend 16 hours in front of your computer at home?

25. Do you own more than $2000 worth of "home computer" equipment?

26. Do you write a "quick and dirty" chequebook balancing program when you pay the monthly bills, because it is too much trouble to find a calculator?

27. Do your kids say things like, "Compile error Dad, so you cannot mow the lawn today.", "I got a priority interrupt and have to go to the library and process some homework!"?

28. Do you ever confuse computer terms with cliches like: "Man, he did he ever blow his stack pointer!"

29. Do you print the kids' school valentines, Christmas, or birthday cards with the computer?

30. Do you constantly lose important phone numbers and info under piles of computer manuals and printer paper?

31. Do you have a neurotic fear of throwing away any box that once contained computer equipment, in case you might have to "send it back to the factory"?

32. Do you have piles and piles of such boxes, filling every closet and every inch of garage space in your living area?

33. Do you have to take out several trash bags full of obsolete program listings every week?

34. Does your spouse often threaten your computer with violence... in a "joking" way, of course? (Examples: "I will take an axe to that thing!", "throw it in the pool", etc.)

35. Have YOU ever threatened a guest with violence if he set a glass of water down next to the computer?

36. Do you talk to the computer as if it could hear you?

37. Do you own more than two computer languages that you never use? And never even learned?

38. Do you own a shirt-pocket pencil holder?

39. When your girlfriend (wife, etc.) says "it is too hot, I think I will slip into something more comfortable", do you run to turn up the air conditioner to protect your computer from overheating?

40. Do your family and friends write you letters instead of calling since they can never get through to you on the phone, while you play on Texpac BBS and the like?

41. Are all the clocks in your house 24 hour format?

42. Do you put Audio CDs in your computer's CD-ROM player to analyze them?

43. Do you take your family on a "get away" vacation to Silicon Valley?

44. Have you worn the letters off your computer's keyboard?

45. Can you recite the alphabet in ASCII codes?

46. Has your dog ever attacked, or raised its leg upon, your computer system out of jealousy?

47. When your wife says she is going to take a "drive to the store" do you get a sudden urge to go check on your disk drives?

48. Is your idea of a BIG ADVENTURE playing one?

49. Can your children do binary arithmetic?

50. Have burglars ever hit the entire neighbourhood except your house, because you are always up programming?

51. Is the biggest tragedy in your life a power surge?

52. Do you use computer chips instead of thumb tacks?

53. Have you actually ever managed to finish a programming project on time?

54. Have you ever realized that you "forgot to go to bed" when the alarm goes off, while sitting in front of the keyboard?

55. Have you ever gotten a new toy for your child that requires "parental assembly" and gone to fire up your Editor Assembler?

56. Do you program in ASSEMBLY LANGUAGE rather than Pascal, Forth or Extended Basic, just for the FUN OF IT?

57. Do your neighbours call the air force because of a strange glow emanating from one room in your house all night long?

58. Do strange noises, frequent orders for chinese food and many packages from the USA (which you eagerly waiting for) catch their attention?

59. Do you celebrate TI-FARE as a national holiday?

60. Can you hotwire a phone into your direct connect modem from any motel or relatives house?

61. Do you travel with a computer?

62. Have you forgotten how to talk to other parents at the P&C meeting?

63. Do you burst out laughing when your spouse is talking and manages to make a completely hidden reference outside of the context of the subject which is hilarious when applied as a computer joke?

64. Does a newly discovered BBS become a highlight of your day? (Depending upon if you are a member of Texpac.)

65. Do you string your own telephone wires and electrical extensions?

66. Have you ever had to explain to the Telecom why you need 4 phones?

67. Are you constantly trying to find a "better disk editor" or a "better input routine" or a "better word processor"???

68. Have you ever finished a program and delivered it, then never modified it again?

69. Do your kids know how to spell RUN before they knew how to spell their own names?

70. Have you noticed how old friends just cannot carry on interesting conversations any more?

I hate to tell you folks, I fell into a lot of these categories, and I am sure you do too! If you have any more of these questions that we could ask, drop me a line and I will write another article, or WRITE one YOURSELF.... GO FOR IT!!! o

continued from page 6

March '89: 80-column analog RGB monitors
August '89: Another approach to a full-screen FORTH editor (4 colours)
November '89: High resolution graphics in FORTH, part 1
February '90: High resolution graphics in FORTH, part 2
May '90: More on high-resolution graphics o

continued from page 13

```
100 ! LABEL
110 ! Extended Basic V3.0
120 ! By Jim Swedlow
130 ! 22 Mar 86
140 ! Based on a program by
      Phil Barnes
150 GOTO 180 :: IN$,SS$,DS$,PI$ :: DIM Y$(4),A$(5):: C$,
D$ :: DIM F$(144),T$(144):: B,J,D,E$,I,C,@ :: CALL KEY
160 DATA "       FIRST CLASS","     MAGNETIC  MEDIA",DO
NOT BEND * DO NOT X-RAY,DO NOT EXPOSE TO MAGNETISM,
170 DATA Your Name,Street Address,"City, CA ZIPCD",
180 @=1 :: CALL CLEAR :: FOR J=0 TO 14 :: CALL COLOR(J,1
6,@):: NEXT J :: CALL SCREEN(5):: !@P-
190 E$=CHR$(27)! ESCAPE
200 DS$=E$&"G" ! DOUBLE STRIKE
210 C$=CHR$(15)!  CONDENSED
220 D$=CHR$(14)!  DBL WIDTH
230 PI$=CHR$(18)! PICA
240 SS$=E$&"S"&CHR$(0)! SUPERSCRIPT
250 IN$=E$&"@" ! INITIALIZE PRINTER
260 E$=E$&"3"&CHR$(12)! CHANGE LINE FEED TO 12/144 INCH
270 PI$=PI$&C$&D$ :: DS$=DS$&C$ :: SS$=SS$&E$ :: Y$(@)="
DF" :: Y$(2)="DV" :: Y$(3)="IF" :: Y$(4)="IV" :: D$="DSK
1" :: C$=""
280 OPEN #1:"PIO" :: DISPLAY AT(3,9):"LABEL 3.0": :"Che
ck the position of the  labels before printing."
290 DISPLAY AT(20,@)BEEP:"<A>ddress":"<C>ustom ":"<D>isk
":"<W>arning"
300 GOSUB 550 :: IF I>4 THEN 300 ELSE B=@ :: GOTO 330
310 DISPLAY AT(10,@): : :"  Labels/Disk: ";B:"     Dr
ive:     ";D$:"     Comment:":TAB(6);C$: :
320 GOSUB 540
330 ON I GOTO 470,500,310,470,370,340,600
340 ACCEPT AT(12,20)SIZE(-2)VALIDATE(DIGIT)BEEP:E$ :: IF
E$="" THEN 340 ELSE B=VAL(E$):: IF B=0 THEN 340
350 ACCEPT AT(13,20)SIZE(-@)VALIDATE("12")BEEP:E$ :: IF
E$="" THEN 350 ELSE D$="DSK"&E$
360 ACCEPT AT(15,6)SIZE(-25)BEEP:C$ :: GOTO 320
370 ON ERROR 580 :: C=0 :: DISPLAY AT(10,@):: DISPLAY AT
(20,@):"Initializing" :: OPEN #2:D$&".",INPUT ,RELATIVE,
INTERNAL
380 INPUT #2:F$(C),I,I,I :: T$(C)="FREE "&STR$(I)
390 DISPLAY AT(22,@):F$(C);"";T$(C):: IF C=127 THEN 42
0 ELSE INPUT #2:F$(C+@),I,J,J
400 IF F$(C+@)="" THEN IF C>5 THEN 420 ELSE C=C+@ :: GOT
O 390
410 I=ABS(I):: C=C+@ :: IF I=5 THEN T$(C)="Prog" :: GOTO
390 ELSE T$(C)=Y$(I)&STR$(J):: GOTO 390
420 CLOSE #2 :: ON ERROR STOP :: DISPLAY AT(20,@):"Print
ing": : : :: FOR B=@ TO B :: J=0 :: D=8 :: E$=" "
430 PRINT #@:DS$;F$(0);E$;C$;E$;T$(0);SS$: :
440 FOR I=J+@ TO J+D :: PRINT #@:F$(I);TAB(12);T$(I);TAB
(18);F$(I+D);TAB(29);T$(I+D);TAB(35);F$(I+2*D);TAB(46);T
$(I+2*D):: NEXT I
450 J=J+24-6*(D=10):: IF C>J THEN D=10 :: PRINT #@: : :
:: GOTO 440 ELSE PRINT #@:IN$
460 NEXT B :: B=B-@ :: FOR I=@ TO C :: T$(I),F$(I)="" ::
 NEXT I :: GOTO 320
470 IF I=4 THEN RESTORE ELSE RESTORE 170
480 FOR C=@ TO 5 :: READ A$(C):: NEXT C :: D=10 :: GOSUB
 570
490 GOSUB 540 :: ON I GOTO 470,500,310,470,530,510,600
500 FOR C=@ TO 5 :: A$(C)="" :: NEXT C :: D=@ :: GOSUB 5
70
510 FOR C=@ TO 5 :: ACCEPT AT(C+9,2)SIZE(-28)BEEP:A$(C):
: NEXT C
520 ACCEPT AT(16,19)SIZE(-2)VALIDATE(DIGIT)BEEP:E$ :: IF
E$="" THEN 520 ELSE D=VAL(E):: IF D THEN 490 ELSE 520
530 DISPLAY AT(20,@):"Printing" :: FOR I=@ TO D :: FOR C
=@ TO 5 :: PRINT #@:PI$;A$(C):: NEXT C :: PRINT #@:IN$:
:: NEXT I :: GOTO 490
540 DISPLAY AT(20,@)BEEP:"<A>ddress <P>rint labels":"<C
>ustom <M>odify defaults":"<D>isk <Q>uit":"<W>arning"
550 CALL KEY(3,I,C):: IF C<@THEN 550 ELSE IF I=13 OR I=
32 THEN I=80
560 I=POS("ACDWPMQ",CHR$(I),@):: IF I THEN CALL KEY(5,C,
C):: IF B THEN DISPLAY AT(20,@): : : : : :: RETURN ELSE
RETURN ELSE 550
570 FOR C=@ TO 5 :: DISPLAY AT(C+9,@):">";A$(C):: NEXT C
:: DISPLAY AT(15,@): :"How many labels: ";D :: RETURN
580 DISPLAY AT(10,@)BEEP:D$;" Could not be accessed" ::
ON ERROR 590 :: CLOSE #2
590 ON ERROR STOP :: RETURN 320
600 CALL CLEAR :: CLOSE #@ :: END                    o
```

Beginning Forth - part 18

by Earl Raguse, UGOC, CA USA

VECTORS, CONSTANTS, VARIABLES AND ARRAYS This time I will use Forth's powerful extensibility capability via the words <BUILDS ... DOES> ... to create a defining word which in turn can define an array variable. I discuss vectoring, a powerful feature of Forth for which I have not yet found much use, but which could solve problems that are otherwise intractable. Vectoring is very similar, if not identical, to indirect addressing which is one of the standard addressing modes available in Assembly Language. Indirect addressing, including auto incrementing, is also available in the Forth Assembler, but that is another story for a later date.

CONSTANTS, VARIABLES AND ARRAYS

There are times that one needs a method to store, manipulate, and keep track of a series of related numbers or strings. A convenient way is to define an array. In BASIC we call this a Subscripted Variable. Arrays may be uni-dimensional or multi-dimensional. A uni-dimensional array is just a list. A two-dimensional array is a table with two or more columns (lists). A multi-dimensional array is a set of two or more of two-dimensional tables, and so on.

First I will show how another defining word (Name) can be created with Name <BUILDS ... DOES> ... which is how the word CONSTANT was defined, CONSTANT is a defining word which in turn is capable of creating an entity (constant) which stores and delivers a constant value.

I am sorry that I cannot explain in detail how all this works, but in general <BUILDS puts Name in the dictionary and executes the words between <BUILDS and DOES> during compile time. The words following DOES> are executed when Name is executed. I understand the simple words which are compiled by <BUILDS, for example the word CONSTANT, whose definition is:

```
: CONSTANT <BUILDS , DOES> @ ;
```

It is used as follows, to define a constant named JUNK.

```
89 CONSTANT JUNK
```

<BUILDS causes CONSTANT to compile the name JUNK into the dictionary, then the , (comma) compiles a value 89 off the stack into memory. DOES> then causes JUNK, when executed, to fetch the compiled value 89 to the stack.

The principle is easy, but knowing what to put between <BUILDS and DOES> and after DOES> for complex words is what makes a good Forth programmer. We will now create ARRAY which in turn will be capable of defining an array name of a size and name we specify. 'Name' will simply become the address of the beginning of the data structure. The definition of ARRAY is as follows:

```
: ARRAY <BUILDS DUP , 2 * ALLOT
          DOES> SWAP 2 * + ;
```

ARRAY is to be used with this syntax:

```
n ARRAY Name
```

This causes a data structure of n*2+2 bytes to be alloted and Name compiled into the dictionary, which when executed, will push its address onto the stack. Only space has been set aside; it is not initialized, except for the value of n which is stored in the first two bytes. I will discuss what the words following DOES> do later. We use ARRAY like this.

```
10 TENNUMB ARRAY
```

defines an array called TENNUMB whose length is adequate for 10 values. To insure that it is all zeros we may use the word FILL as follows.

```
TENNUMB 2+ 10 2 * 0 FILL
```

Where TENNUMB is the address, 2+ jumps over the length value 10, 10 is for ten values, 2 is for two bytes each and 0 is how we want them initialized. We could, of course, replace 10 2 * with 20, and I do that on Screen #29. To later store a value 150 in say the seventh position in our TENNUMB array, we write

```
150 7 TENNUMB !
```

What happens here is a function of what followed DOES> in the above definition of ARRAY. Recall that that was SWAP 2 * +. TENNUMB pushes its address onto the stack, SWAP exchanges the address with 7, 2 * makes that 14 and + adds that to address to offset to the actual address of the seventh position, ! then takes the 150 off the stack and stores it there. To retrieve the value, we simply write

```
7 TENNUMB @
```

Notice that this is almost identical to single variables defined with VARIABLE, except that we must supply the subscript to indicate which of the multiple values, stored in the array, that we want. Isn't that easy? You do not even have to put the subscript in parentheses like in BASIC. Of course the subscript can be another variable, even another array value. Typically it will be the index I of a DO...LOOP.

Screen #29 has all the above ARRAY words but I have not included any application examples, I leave that to you.

Jenny's Younger Set

Here is a program from my friend Vincent Maker. At least he writes to me!

```
100 REM A DOCTOR WHO QUIZ
110 REM BY V.  MAKER
120 REM FOR MELANIE
130 CALL CLEAR
140 PRINT "1. WHO HELPED THE DOCTOR IN DEATH TO THE
    DALEKS?"
150 PRINT "A) BELALL"
160 PRINT "B) THE DALEKS"
170 PRINT "C) THE EXILLIONS"
180 PRINT "PRESS YOUR GUESS"
190 CALL KEY(0,K,L)
200 IF L=0 THEN 190
210 IF K<>65 THEN 240
220 RIGHT=1
230 GOTO 250
240 WRONG=1
250 PRINT "2. WHO BETRAYED KING PELADON IN CURSE OF
    PELADON?"
260 PRINT "A) HEPESH"
270 PRINT "B) ISLXYR"
280 PRINT "C) SSORG"
290 CALL KEY(0,J,K)
300 IF K=0 THEN 290
310 IF K=65 THEN 330
320 WRONG=WRONG+1
325 GOTO 340
330 RIGHT=RIGHT+1
340 PRINT "3. ON WHICH PLANET DID THE DALEKS SEARCH FOR
    THEIR DESTINY?"
350 PRINT "A) SKARO"
360 PRINT "B) SPIRIDON"
370 PRINT "C) EARTH"
380 CALL KEY(0,J,M)
390 IF M=0 THEN 380
400 IF J=65 THEN 430
410 WRONG=WRONG+1
420 GOTO 450
430 RIGHT=RIGHT+1
```

continued on page 4

Writing in Machine Code

Setting memory to >FF

by J.E. Banfield

"Love is not for the faint at heart"

Warning and disclaimer: Because machine code programs can execute everything that the hardware is capable of doing and there are no inbuilt checks and safeguards, a silly instruction is capable of devastating the system and of destroying data in RAMs, RAMdisks and even (unlikely) may corrupt data on disk. You may well create your very own computer virus. Accordingly, turn off any peripherals which may contain valuable data (including of course programs), disconnect your PEBox and save any material in MiniMemory before typing in my, or your own machine code routines.

For example, should you execute the two word code:

```
04 F5
10 FE
```

you will set to zero (clear) all data in RAMs in addresses up to the location of these instructions, which will themselves be contaminated. Similarly:

```
07 35
10 FE
```

if executed will set memory to all ones, that is, FF FF ...

To see this we examine:

```
0 0 0 0   0 1 1 1   0 0 1 1   0 1 0 1
\_____070_____/ \_____/
   op-code        S address
```

From Figure 1 (TND, June 1992, page 14), 070 or SETOS translates to 'set ones to source'. The S address has address mode '3' and refers to Ac 5 (R5); take the source address from Ac 5 and then increment it; in this case by 2. The next instruction 10 FE is a skip always instruction with a count of minus 2; it causes the instruction executed next to be two instructions above the following instruction; namely it executes the 07 35 instruction again in an endless loop (see last month's contribution). Each time the two instruction loop is executed, a new memory address is set to ones until eventually the first instruction itself is overwritten and other things happen (see below).

If you are brave and do not mind overwriting the current contents of your MiniMemory module and take all other precautions, you might try the following:

Insert MiniMemory, turn on console, press '2' three times and when '?' appears, type in 'M7FC0<enter>' and type in the numbers in quotes after the arrows followed by '<enter>':

```
M7FC0 = XY --> '07'
M7FC2 = XY --> '35'
M7FC4 = XY --> '10'
M7FC6 = XY --> 'FE'
```

Type in a '.' to return to the '?' prompt to allow the contents of memory to be examined. Start at 'M7FB0' to look at consecutive memory locations by pressing the '<space>' bar, checking the validity of the program as you go. Then take a deep breath, say your prayers and after typing '.' to display '?', type 'E7FC0<enter>'. Nothing will appear to happen. You have entered an infinite loop.

Switch off the console and then on again. Press '2' three times to bring up the '?' and examine 'M7000' ... as above. When you tire of pressing the '<space>' bar, type '.', 'M7FB0<enter>' and continue to step through the memory, when you will reach:

```
M7FC0 = FF -->
M7FC1 = FF -->
M7FC2 = 10 -->
M7FC3 = FE -->
```

with further memory containing the original contents. I have selected Ac 5 in this example since it contains 02 70 at execution start.

Now change the program to:

```
M7FC0 = FF --> 04
M7FC1 = FF --> F5
M7FC2 = 10 -->
```

Check your program as above. The first instruction now reads:

```
0 0 0 0   0 1 0 0   1 1 1 1   0 1 0 1
\_____04C_____/ \_____/
   op-code        S address
```

The op-code translates to SETZS (see Figure 1 mentioned earlier), which is set source contents to zero. Now press '.' and type 'E7FC0<enter>' and you again enter the infinite loop. Switch off and on again and you will find that memory previously filled with FF FF will now contain 00 00. Repeat the first program if desired.

Perhaps these instructions are special and unlikely to be encountered. I am indebted to Emeritus Professor Robin Stokes, who built a 9900 based 16 bit data bus computer in the 1970s (a machine more powerful than an extended TI99/4A) for this caution in executing a block of FF FF "instructions" which tend to appear in partially written EPROMs. This can cause untold woes and it is easy to see why. A single FF FF instruction is:

```
1 1 1 1   1 1 1 1   1 1 1 1   1 1 1 1
\_F_/ \_____/ \_____/
  op-code   D address   S address
```

The op-code F corresponds to IOR byte, which is inclusive OR byte, which sets a bit if the corresponding bit in either source or destination is a 1. The source address @F+ means take the source address from Ac F (R15) and then increment it (by one since it is a byte instruction). The destination address, the new contents of Ac F is used and then incremented. Accordingly the contents of each second memory location are corrupted each time an FF FF instruction in the block is executed. Strange things do happen! o

continued from page 16

The other more useful procedure to consider is SEG$. This returns a substring of the input string. It takes as inputs a string, the position in the string where the sub-string starts and the length of the sub-string required. If the position is outside the range of the string a null string is returned. If the position plus the length is outside the range of the input string, only the characters to the end of the string are returned. As in the previous case, C cannot return a string so we will have to have another parameter in the procedure definition to take the output sub-string. The function call could be segment(s1, s2, pos, len), with s2 being a pointer to the input string, pos an integer with the starting position in the input string and len an integer with the length of the sub-string. s1 is a pointer to the output string. Consider the following attempt at this procedure.

```
segment(s1, s2, pos, len)
char *s1, *s2;
int pos, len;
{
  int i, n;
  n = strlen(s2);
  if (pos <= n && pos > 0) {
    if (pos > 1) {
      for (i = 1; i < pos; i++, s2++) ;
    }
    strncopy(s1, s2, len);
  }
  else
    *s1 = NULL;
  return;
}
```

This is easy to do using the strncpy() procedure. First the pointer to s2 must be adjusted to point to the start of the requested sub-string and then strncpy() does the rest. Some checking of the inputs is done first using the AND function of C (two ampersands).

Well that is all I have time for this month. I have not tried out the last two procedures and perhaps next month I will look at writing a test program for these and the others from this month. Have a go yourself and you will learn so much faster. Remember, if you want to get some advice, give me a ring or write me a letter. o

10.14.12

Regional Group Reports

Meeting Summary For AUGUST

Banana Coast	09/08/92	Sawtell
Central Coast	08/08/92	Saratoga
Glebe	06/08/92	Glebe
Hunter Valley	08/08/92	
Illawarra	10/08/92	Keiraville
Liverpool	07/08/92	
Northern Suburbs	27/08/92	
Sutherland	21/08/92	Jannali

BANANA COAST Regional Group
(Coffs Harbour Environs)

We never miss meeting at Kerry Harrison's residence 15 Scarba St. Coffs Harbour, 2 pm second Sunday of the month. Visitors are most welcome. Contact Kerry 52 3736, Kevin 53 2649, Rex 51 2485 or John 54 1451.

CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043) 69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce Street, Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The meetings are usually held on the second Saturday of each month at members homes starting at 3:15 pm. Check the location with Geoff Phillips on (049) 428 176. Note that after 9:00 pm this number is used for the ZZAP BBS which includes TI-99 information. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the second Monday of each month after the TIsHUG Sydney meeting, except January, at 7.30pm, at the home of Geoff & Heather Trott, 20 Robsons Road, Keiraville. A variety of activities accompany our meetings, including Word Processing, Spreadsheets and hardware repairs. Contact Lou Amadio on (042) 28 4906 for more information.

LIVERPOOL Regional Group

Regular meeting date is the Friday following the TIsHUG Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 6447377 (home) or (02) 7598441 (work) for more information.

NORTHERN SUBURBS Regional Group

Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02) 918 8132. Come and join in our fun.

Dick Warburton.

SUTHERLAND Regional Group

It has been a few months since my last Regional report but the group has been meeting on a regular basis, each month. June was no exception, with a good roll up in attendance.

Herbert Schade demonstrated some of his many TI Artist instances, all of which have been catalogued in hard copy form.

Derek Wilkinson entertained the group with his Tigercub programable calculator, which is a very good piece of software. A review was contained in the July edition of TIsHUG.

We also spent some time in resurrecting deleted files using Disk Utilities. A good time was had by all.

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm.

Peter Young

The cut-off dates for submitting articles to the Editor for the TND via the BBS or otherwise are:

September	9 August
October	13 September

These dates are all Sundays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00pm, at the latest. Longer articles should be to hand well before the aboves dates to ensure there is time to edit them.

****

TIsHUG in Sydney

Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are held at the RYDE INFANTS SCHOOL, Tucker Street (Post Office end), Ryde. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

August

New software and hardware to be demonstrated.

September

The second all day tutorial session. Your chance to learn about using software, writing programs or understanding hardware. We can provide anything that you want but you must tell us what you would like and at what level you would like it.

October

The third buy, swap and sell day. This one is in the middle of the school holidays but plan to take the day off and see what is about.

November

The TI-Faire will be a few weeks after this meeting so it may be taken up with the organizational requirements of this big day. New software and hardware to be demonstrated. Watch this space for more details. Time to think about nominating for positions on the board. I am sure there will be some vacancies this year!

December

The Annual General Meeting followed by some festive eats and drinks. There will probably be a bit of celebration after the TI-Faire, if we are all still friendly after the event. Make sure that you attend and give your support to all the workers in the club. O

continued from page 12

With this method, you can print individual lines or words in italics, double struck, underlined, superscript, emphasized, or in different NLQ fonts or different colors. However, do not use any CTRL U codes for a feature that you plan to select from Printall, or you will turn it off for the rest of the text.

NOTE: When a line contains CTRL U codes, the program will NOT warn you or truncate a line which is more than the selected column width.

Although this program is intended primarily for multiple-column printing, it has other uses. If your letter turns out to be 70 lines long and you would like to print it on one page, use this program and select 70 lines. If you need a double-spaced manuscrpt, select 30 lines. If you need a tiny list, such as a list of the songs to put in the case of a music cassette, select elite condensed superscript and 120 lines per page.

Since the TI99/4A can only store strings of about 12.5K of console memory may get a MEMORY FULL error if you try try to format much more than 60 lines of condensed print per page. You can gain an extra 1036 bytes by entering a CALL FILES(1) and then NEW before loading this program. O