

TI HARDWARE AND COMPONENTS

CABLES	£	p
DISK DRIVE CABLE EXT X 2	08.00	
DISK DRIVE CABLE EXT X 2 LONG	09.00	
DISK DRIVE CABLE INT S/H	02.50	
DISK DRIVE CABLE INT X 2	08.00	
DISK DRIVE CABLE EXT X 2	08.00	
DISK DRIVE POWER SPLITTER	05.50	
PARALLEL PRINTER CABLES	08.50	
COMPONENTS		
DISK CONTROLLER CONNECTORS 34 WAY	01.25	
DISK DRIVE CONNECTORS 34 WAY IDC	01.10	
DISK DRIVE POWER CONNECTORS IN or EX	01.00	
9 PIN "D" CONNECTOR & COVER	01.00	
PARALLEL PRINTER CONNECTORS	03.00	
16 PIN PRINTER CONNECTORS	01.00	
P.E.B PUSH BUTTON ON/OFF SWITCH HEAVY DUTY REPLACEMENT	01.75	
PROGRAM HEADERS 8 WAY (FOR SETTING DISK DRIVES)	00.50	
SHALL BLACK PLASTIC BOXES 72x45x22 mm	01.00	
16 way RIBBON CABLE IDC PER FOOT	00.50	
1N4148 TYPE DIODES FOR JOYSTICKS ETC	00.05	
DISKS AND ANCILLIARIES		
50 CAP DISK BOX WITH LOCK	07.50	
100 CAP DISK BOX	10.50	
140 CAP DISK BOX	12.50	
5.25 DSDD DISKS 96tpi 100% ERROR FREE		
LIFETIME GUARANTEE		
25+ 50+ 75+ 100+		
36p 34p 32p 30p		

HARDWARE	NON MEMBERS	MEMBERS
JOYSTICK INTERFACE TI to 2 ATARI	09.50	07.00
CONSOLES RECONDITIONED	25.00	20.00
DISK DRIVES INT SS.FH 40 TRACK	35.00	25.00
POWER SUPPLIES	10.00	07.50
MODULATORS	10.00	07.50
JOYSTICKS TI	10.00	07.50
CASSETTE LEADS	07.50	05.00
PAR PRINTER LEAD	12.00	08.50
ALPHA RS232 INTERFACE	07.50	05.00
MINIMEMORY BATTERIES	01.25	01.00

CARRIAGE as previous list.  
 DISK BOXES 02.00p  
 Cheques etc Made out to M.Goddard  
 Overseas cheque or money order drawn on London  
 Overseas carriage at cost call for details

MIKE GODDARD COMPUTER SUPPORT TEL 0978 843547  
 "SARNIA" CEMETERY RD, RHOS, WREXHAM, CLWYD.

# TI\*MES

TI99/4A USERS GROUP (U.K.) CONTACTS LIST:

Chairman: Gordon Pitt: (Tel 0922 476373)  
 259 Sneyd Lane, Bloxwich, WALSALL, W Midlands, WS3 2LS  
 Treasurer: Alan Rutherford (Tel 0625 524642. Bus: 061 223 1301 Ext2198)  
 13 The Circuit, WILMSLOW, Cheshire, SK9 6DA.  
 Publicity Officer: Christine Bennett. (Tel 061 430 7298)  
 20 Oak Avenue, Romiley, STOCKPORT, Ches (Bus: 061 430 7645)  
 Membership Secretary and Telecoms: Peter Walker:  
 24 Bacons Drive, CUFFLEY, Herts, EN6 4DU. 0707 873778  
 General Secretary: Jim Ballinger:  
 5 Offerton Avenue, DERBY, DE3 8DU. 0332 772612  
 TI\*MES Editor: Alan Bailey:  
 14 Shelley Grove, LOUGHTON, Essex, IG10 1BY. 01 508 1053  
 LIBRARIANS:  
 Cassette: Tim Anderson (Tel 041 337 1291. Bus 041 339 9627):  
 Ground Floor (Left), 47 Apsley Street, GLASGOW, G11 7SN.  
 Disk: Stephen Shaw (also Journal Exchange & V.P.)  
 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH.  
 Hardware (& DIY Projects): Mike Goddard (Tel 0978 843547):  
 Sarnia, Cemetary Road, Rhos, WREXHAM, Clwyd, LL14 2BY.  
 Module: Edward Shaw  
 Crow Holt Farm, Basford, LEEK, Staffs, ST13 7DU. 0538 360382  
 Publications: Mike Curtis (Tel. 0209 219051):  
 21 Treliske Rd. Roseland Gardens, REDRUTH, Cornwall. TR15 1QE.  
 Correspondence relating to general group matters to the General Sec.

Page No.	MAGAZINE CONTENTS
IFC Editorial, Disclaimer	
1 Membership News, Peter Walker	
2 Console Only Corner, Peter Walker	
4 Publications Library Report, Mike Curtis	
6 Hardware Update, Mike Goddard	
7 Cassette Reviews, Nicky Goddard	
8 Chester Workshop, Mike Goddard	
9 Rambles, Stephen Shaw	
19 Disk Library Report, Stephen Shaw	
23 Book Reviews, Stephen Shaw	
25 G, Stephen Shaw	
27 Logo, Stephen Shaw	
29 Treasure Trail, Peter J Hutchison(SJS)	
29 Rambles, Stephen Shaw	
30 Textaments, TI Base Material(SJS)	
31 New Music from Harrison Software, Bruce & Dolores Harrison(SJS)	
36 Tips from Tigercub, edited by Stephen Shaw	
40 Quintus, Sam Pincus(SJS)	
43 Sector Sharing, Mark Shafer(SJS)	
45 Rambles, Stephen Shaw	
47 Telephone Dialler Project, Peter Walker & Mike Goddard	
52 Myarc HFDC Hard Disk Controller, Jan Alexandersson(SJS)	
59 XB, Jim Swedlow(SJS)	
60 Rambles, Stephen Shaw	



## EDITORIAL

Enclosed is a letter from our new Chairman calling for more participation by our members not only in our own activities, but also in others covering the field of Home Computers. We appreciate that our wide geographical distribution makes it more difficult, and expensive, to come to centrally organised meetings, but there are almost certainly other members near you, see Peter Walker's lists, so a local meeting may be a possibility! Why not give it a try?

As you will see from the many excellent contributions, there is a heartening level of activity in many parts of our group. Especially satisfactory are the evidences of continued hardware development, ably led by our own Computer Support run by Mike Goddard. The orphan should be able to look forward to a long life! The Zenoboard sounds like a wonderful idea, and if it really does not need a separate power supply, as is suggested, it should be a very nice modification. Also very interesting is the article from Sweden on disk control. Perhaps we could hope for a contribution from the UK on the disk interface system of the TI99, enlarging on the Swedish information at a more basic level.

Happy computing, Alan.

### DISCLAIMER

The views expressed in this magazine are those of the contributor and do not necessarily represent those of the group or Committee.

NEXT COPY DATE IS 1st. DECEMBER.

TI\*ME

## MEMBERSHIP NEWS from Peter Walker

Since the last issue of TI\*MES we welcome to TIUG(UK): Peter Boswell, Gary Smith, Roger Rocks, Graham Steward, Royston Meredith and David Gale.

In issue 24 I printed a list of members who had agreed to their names being printed in TI\*MES. The object of this is to encourage members in the same district to get together and perhaps form local groups. Here are some further names from the membership list.

### SCOTLAND

TIM ANDERSON	GROUND FLOOR (LEFT), 47 APSLEY STREET, GLASGOW, E11 7SN
ALASDAIR BRYCE	51 DUMBUIE AVENUE, SILVERTON, DUMBARTON, SCOTLAND, G82 2JH
ROY DOUGLASS	BRIDGES' COTTAGE, AROS, ISLE OF MULL, PA72 6JF
P JACKSON	61 CRUMMOCK GARDENS, BEITH, Ayrshire, KA15 2HF
ERNEST KERR	120 SEMPILL AVENUE, ERSKINE, SCOTLAND, PA8 6DB
CHARLIE MCROBERTS	113 REDBURN ESTATE, ALEXANDRIA, DUMBARTONSHIRE, G83 9BT
ROBERT R WILSON	25 MUIREND STREET, KILBIRNIE, Ayrshire, KA25 7DS

+ 1 other.

### WALES & THE MARCHES

ERIC DYKE	1 ABBEYDALE COTTAGE, WOODFIELD, HUXLEY, CHESHIRE, CH3 9BN
MIKE GODDARD	*SARNIA*, CEMETERY ROAD, RHOS, NR WREXHAM, CLWYD, LL14 2BY
ANDREW J LARSEN	121 MAES TY CANOL, BAGLAN, PORT TALBOT, WEST GLAMORGAN, SA12 8UR
KENNETH F WILFORD	27 WYNDHAM ROAD, BLACON, CHESTER, CHESHIRE, CH1 5SE
MARK R WILLS	37 ABBOTS ROAD, MONKMOOR ESTATE, SHREWSBURY, SHROPSHIRE, SY2 5PZ

+ 1 other.

### NORTH-EAST MIDLANDS

JIM BALLINGER	5 OFFERTON AVENUE, DERBY, DE3 6DU
EMMA DAVEY	*KATCHIKALLI*, PLOUGH LANE, FISKERTON, LINCOLN, LN3 4EY
IAN A HEWITT	26 OUTWOODS DRIVE, LOUGHBOROUGH, LEICESTERSHIRE, LE11 3LT
GEORGE MICHEL	34 PIPPIN HILL, DENBY VILLAGE, DERBY, DE5 8PD
SAM WARDLE	83 NORRETT ROAD, ARNOLD, NOTTINGHAM, NG5 8EA

+ 4 others.

### BIRMINGHAM & NORTH-WEST MIDLANDS

ROBERT J BATES	34 BROWNS LANE, TAMWORTH, STAFFORDSHIRE, B79 8TF
ROGER HAWKES	THE FAIRWAYS, 65 ST ANDREWS ROAD, SUTTON COLDFIELD, WEST MIDLANDS, B75 6UH
ROYSTON H MEREDITH	104 BOURNBROOK ROAD, BIRMINGHAM, B29 7BU
EDWARD H SHAW	CROW HOLT FARM, BASFORD, LEEK, STAFFORDSHIRE, ST13 7DU
ERIC J WILLIAMS	211 MANOR WAY, CREWE, CHESHIRE, CW2 6PH

+ 2 others.

### EASTERN COUNTIES

GEOFFREY P D COAN	39 AVOCET, LETCHWORTH, HERTFORDSHIRE, SG6 4TH
DAVID GORDON	44 LADYSHOT, HARLOW, ESSEX, CM20 3EL
JOE F HUNT	10 OAK LANE, LAKENHEATH, SUFFOLK, IP27 2RJ
CHRISTOPHER T SNELL	6 BELMONDE DRIVE, SPRINGFIELD, CHELMSFORD, ESSEX, CM1 5TY
JOHN STOCKS	11 STONEHILL ROAD, ROXWELL, CHELMSFORD, ESSEX, CM1 4PF

+ 8 others.



September 1989

259 Sneyd Lane  
Bloxwich  
WALSALL  
West Midlands  
WS3 2LS

Tel: (0922) 476373

Dear TIUGUK members,

As your new Chairman, it is my pleasure to thank the outgoing Chairman CLIVE SCALLY on behalf of us all, for his significant effort towards making TIUGUK a going concern from its early beginnings, and I am certain that you would join me in wishing him well in his future endeavours.

The TI community is now served by more groups of dedicated individuals than at any other time in its eleven year history. In the beginning was TIHOME, succeeded by TIHCUC (the less said about that, the better), TI-EXCHANGE (the fore-runner of TIUGUK), and what was OXON TI USERS and is now ITUG (soon to resume its magazine, I understand, and Peter has still been as active as work permits), together with my own group - WMTI USERS, and in recent times, the growing EAR 99ERS under Scott and JoAnn, who have managed what no others have: regular monthly meetings; and other groups like DORTIG who each try or have tried to cater for the needs of the people they have attracted.

In most cases, I know that TI owners generally are members of more than one group, perhaps trying to get that little extra dividend that diversification brings, and one of my aims during my chairmanship will be to bring about closer and better co-operation between groups, and I hope to bring our committee round to my way of thinking in the near future. We can only benefit by strengthening our bonds, and making them more resilient in the face of the changing market. We need to put our heads together to come up with new ideas, greater challenges for those of our readers who enjoy them, as well as consolidating the sources of information for our members who are just taking those first tentative steps out on the road to computer literacy with the TI-99/4A.

We must always look for ways to improve the content of TI\*MES, and to cater as fully as possible for the widest range of Users' experience.

I would like to see three main sections: Games/Unexpanded, TI Expanded, and Third Party Expanded (i.e., Myarc, Corcomp, Mechatronic, etc.), and would expect that each subgroup would have something to learn from another. Of course, that means we need more authors: if you want to read about something in TI\*MES and it never seems to appear, then why not become a writer yourself? This is a clarion call we have all seen in all TI magazines and newsletters; let's see if TI\*MES readers can answer the call!

We must always look for ways to improve and expand the services we offer within TIUGUK, and outside, and I would like to see us become less inward-facing and more outward-looking in our approach.



One way we can do this is to participate more ourselves. One of my jobs in WMTI Users is to organise three Workshops a year (we'll be doing number 7 I think in October) which are not restricted to just WMTI Users (anyone can come, whether they have a TI or not!) and you don't have to have an egg-shaped head to attend. Make the effort to come to at least one a year - you could be in for a pleasant surprise; many visitors have gone away feeling tired but elated, perhaps having found a missing piece of the jigsaw, or having established a contact with a kindred spirit (or should that be sprite ?).

We have our yearly AGM - it's only once a year, but how many of us make the effort to come (and how many criticise but never show up at the AGM to make their voice heard ?) ?

I'm taking a tiny group over to Nijmegen in October - about a week or two before the Workshop - to make a UK presence at a European TI event; I'd like to take a bigger group; you, the reader, can make that a reality.

The Alternative Micro Show is another opportunity to fly the TI flag, although it does rather dilute us; however, just as with the bigger shows like the PCW Show, and the Which ? Computer Show, it is an opportunity to meet up, see what the rest of the world is doing (you never get anywhere by ignoring what others are doing), and if nothing else, it gets you out of the house.

Even if you don't want to get out of the house, you can still keep in touch in other ways. The WMTI Users BBS is now ready to go online (for the moment, by pre-arrangement until a pattern is established), and it is certainly not as cheap as writing, but it is more fun and like credit cards, it is up to you to control the amount of time and money you spend on it.

My joint theme for this year, then, is Co-Operation and Participation, and it all depends on your co-operation and your participation.

Finally, I would like to express publicly my thanks to a tiny bunch of generally unsung heroes whose presence is rarely acknowledged. They provide the backbone for our community, for without them and their assistance, almost all of the groups would not have had hardware repair services, software troubleshooting, and informed advice (discussion of esoteric things like the TI Operating System, GPL, and so on) as well as invaluable contributions to various projects throughout the years.

Naming no names (to avoid embarrassment, among other things) you know who you are, even if modesty might make you disagree, but in any case, we all thank you for your unstinting efforts to keep the rest of us less well-informed TI Users afloat.

I look forward to hearing from and meeting as many people as possible in the coming year,

Gordon Pitt  
Chairman, TIUGUK



By Peter Walker

In this issue I thought we might explore conversions. In many programs one has a need to convert a variable from one form to another. First let's look at the simple conversion functions provided in Basic.

STR\$

This converts a number to its string text equivalent. I find this useful with data storage of short numbers. For example, ignoring leading length indicators, numbers in TI99/4A require 8 bytes of storage while in string form only the number of characters is needed. Eg the number 345 requires 8 bytes of storage while the text "345" only requires 3. STR\$ would also be used if you wanted to concatenate a group of words and numbers together: the numbers must be converted to strings first. For example:

```
A$="The result was "&STR$(N)&" people from a total of "&STR$(M)
```

Note the blank space after 'was' and before 'people'. Unlike when printing numbers, string-form numbers have no trailing or leading spaces.

VAL

This is the opposite of STR\$. It creates the number equivalent to the string-form number. An example of its use might be as follows:

```
100 INPUT "CAR REG NO?":A$
110 N$=SEG$(A$,2,3)
120 N=VAL(N$)
```

In this example a number is extracted in string form from the mixed alphanumeric A\$ car registration number. The numeric part is extracted at line 110 and converted to a number at line 120.

ASC

This is another very useful conversion function. It converts the first character of a string to its equivalent ASCII code. A word of warning: ASC does not like to be presented with a null string as input. If this might occur then use the following:

```
100 IF A$="" THEN 200
110 A=ASC(A$)
```

In this way, null strings are trapped.

CHR\$

This is in effect the opposite of ASC. It converts a ASCII code to the equivalent character in string form. Here is an example of ASC and CHR\$ in use in a routine to return the lower case character of an upper case input.

```
100 INPUT "UPPERCASE CHARACTER":UC$
110 N=ASC(UC$)
120 LC$=CHR$(N+32)
```

NUMBER TO NAME

Now for something a little more complex. You want to convert the numbers 1 to 13 the names of the cards: "ACE", "TWO"... "TEN", "JACK", "QUEEN" and "KING". This is where you would use an array:

```
100 DIM CN$(13)
110 FOR C=1 TO 13
120 READ CN$(C)
130 NEXT C
140 DATA ACE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
150 DATA JACK,QUEEN,KING
160 INPUT "NUMBER? ":C
170 PRINT CN$(C)
180 GOTO 160
```

Note that since the READ is expecting a string input the data doesn't need quote marks "".

Now how would we cope with the reverse situation? We have the text "ACE" or "FOUR" and wish to translate to digits 1 and 4. One way could be to use the array we have created above:

```
100 INPUT "CARD NAME ":N$
110 FOR C=1 TO 13
120 IF N$=CN$(C) THEN 160
130 NEXT C
140 PRINT "NO SUCH CARD"
150 GOTO 100
160 PRINT "CARD NO IS";C
```

The only snag is speed. It takes a while to compare each name with the given text. Is there a quicker method? The answer is yes. We can use that very versatile POS function which quickly searches a string for the presence of a given string. In this case we note that the card names are all different in their first two characters. Now look at this:

```
100 B$="ACTWTHFOFISISEEINITEJAQUKI"
110 INPUT "CARD NAME ":N$ 120 M$=SEG$(N$,1,2) 130 P=POS(B$,M$,1)
140 IF P=0 THEN 170
150 PRINT "CARD NO IS";(P+1)/2
160 ...
170 PRINT "NO SUCH CARD"
```



PUBLICATIONS LIBRARY REPORT #1

\*\*\*\*\*

UNDER NEW MANAGEMENT!

21 TRELISKE ROAD,  
ROSELAND GARDENS,  
REDRUTH,  
CORNWALL,  
TR15 1QE  
TEL (0209) 219051

HI, Having taken over the Publications Library, I received a rather large box of books to sort through. I have now got myself a bit more organized with all of the books sorted out and just a bustin' their spines to get loaned out (TO YOU!) So COME ON! I have got my ready eyes on masses of TI books and am doing my best to get them for the library. ALL library money will buy NEW books for us all.

IF you don't want to loan a book, do you have any module manuals that you don't need to look at any more? If so, would you donate it to the library? With so many programs available on disk (AND cassette!) every bit of documentation we can pass on to a group member maintains his/her interest in the TI. I have had programs from the disk library and helpful as he is, Stephen can't remember every byte or word concerned with a specific program. If you are willing to donate a manual please contact me. \*\* ANY BOOKS NOT IN THIS LIST THAT YOU ARE WILLING TO SELL TO THE LIBRARY CALL ME

LIB.NO.	- TITLE -	- AUTHOR -
TPL001	SOFTWARE DEVELOPEMENT	VINCENT/GILL
TPL002	EDITOR ASSEMBLER MANUAL	TI STAFF
TPL003	TI UESER REFERENCE GUIDE	TI STAFF
TPL004	TI USER READ THIS FIRST	TI STAFF
TPL005	USING & PROGRAMMING THE TI	FREDERICK HOLTZ
TPL006	TI FAVORITE PROGS EXPLAINED	DONALD S KREUTNER
TPL007	YOUR FIRST TI99/4A PROGRAM	RODNAY ZAKS
TPL008	BEGINNERS BASIC	TI STAFF
TPL009	TI WRITER MANUAL	TI STAFF
TPL010	TI WRITER TRICKS & TIPS	JOYCE CORKER
TPL011	INTRO INTO ASSEMBLY LANGUAGE	RALPH MOLESWORTH
TPL012	TI TECHNICAL INFO MANUAL	unknown
TPL013	PROGRAMMING BASIC	HERBERT D FECKHAM
TPL014	GET MORE FROM THE TI	GARRY MARSHALL
TPL015	MASTERING THE TI	PETER BROOKS
TPL016	GETTING STARTED WITH THE TI	STEPHEN SHAW
TPL017	LEARNING TO USE THE TI	KEVIN TOWNSEND
TPL018	GAMES FOR YOUR TI	ANDREW NELSON
TPL019	THE ORPHAN CHRONICALS	RONALD D ALBRIGHT
TPL020	SMART PROGRAMMING GUIDE FOR SPRITES	CRAIG MILLER
TPL021	ENTERTAINMENT GAMES IN BASIC & EXTENDED BASIC (INCLUDES A CASSETTE WITH IT)	KHOA & QUYEN TON
TPL022	THE TEXAS PROGRAM BOOK	VINCE APPS
TPL023	THE BEST OF 99er VOL.1	VARIOUS
TPL024	FUNDAMENTALS OF TI ASSEMBLY LANGUAGE	MORLEY
TPL025	TI MODULE MANUALS	TI STAFF MEMBERS
TPL025	PHYSICAL FITNESS TRAINING	
TPL026	PERSONAL REPORT GENERATOR	
TPL027	PERSONAL RECORD KEEPING	
TPL028	VIDEO CHESS	
TPL029	TOMBSTONE CITY	
TPL030	EXTENDED BASIC REFERENCE CARD	
TPL031	BASIC REFERENCE CARD	

TPL032+ I WOULD LIKE THE LIBRARY TO EXPAND. IF YOU HAVE ANY MODULE MANUALS NOT NEEDED ANY MORE, PLEASE DONATE OR LOAN THEM TO THE LIBRARY.

So, what I am asking is, do you have any documentation you can donate or loan to the library. Lots of these programs are available on disk for us, and some documentation might help if we can get it.

OTHER MANUALS AVAILABLE ARE :-

- FILED 99
  - P.R. BASE
  - T.E 2 PROTOCOL MANUAL (NOT the one with the module!)
  - CREATIVE FILING SYSTEM
  - FORTH MANUAL
  - SUPERBUG II (Version 1)
  - EXTENDED DISPLAY PACKAGE
  - STAR MANUAL
  - DM1002 MANUAL
  - G MANUAL
  - LINKER MANUAL
- Lots of other thinner manuals also.

Other books available (NOT TI!) are :-

1. BBC Users Guide
2. Introduction To Microcomputer Programming
3. ZX SPECTRUM Explored
4. SPECTRUM Users Guide
5. CBM PET Users guide
6. BBC Graphs And Charts

These books I have included as they might be handy for anybody interested in converting from the above to TI basic. I have used them for converting programs from Home Computing Weekly and Personal Computing Today etc. Well that is the state so far, I am compiling a listing of short reviews of each book as a guide to its contents. if you would like a copy of said listing, send an S.A.E. to the address at the start of the article and specify your interest if you want to know about a specific subject.

Bye for now.

Mike Curtis.

contd. from p.3

This use of POS can be very useful in a wide range of applications. If you were writing a program to convert hexadecimal numbers to decimal the following use of POS would give the decimal value of each character:

```
D=POS("1234567890ABCDEF",HEX$,1)
```

**SPEAKING NUMBERS**

A particular use for number to name conversion comes with the Speech Synthesizer" (SS) when speaking numbers. Try this:

```
100 INPUT "NUMBER ":N
110 CALL SAY(STR$(N))
```

This is OK for N=0-9. Above that, however, the SS, if presented with "34" will say "THREE, FOUR" rather than "THIRTYFOUR". The following routine needs to be used to get the SS to speak numbers up to 999.



HARDWARE UPDATE.....

Mike Goddard

I now have some details of the "Zenoboard" this is the expansion board produced in the USA by Eric Zeno. It will allow you to build 32K memory, a battery backed clock and add your extended basic and speech synthesizer to the inside of your console I am hoping to get one as soon as I can get details back from Eric on shipping charges etc. This board I feel will be of equal use to expanded and unexpanded owners alike the basic price of the board is somewhere in the region of £11.00 plus shipping so does look to be a very good buy. I should have details within the next few weeks so if you are interested please get in touch If we can order over 25 boards Eric is offering a discount.

ZENOBOARD SPECIFICATION.

- \* 32K STATIC RAM
- \* BATTERY BACKED CLOCK
- \* SPEECH SYNTHESIZER
- \* EXTENDED BASIC
- \* 3 ADDITIONAL SWITCHED GROM SOCKETS
- \* ANY CIRCUIT CONFIGURATION
- \* NO ADDITIONAL POWER NEEDED
- \* SWITCHES AND LIGHTS CAN BE ADDED FOR ALL OR SOME CIRCUITRY
- \* GROM RESET SWITCH
- \* COMPATIBLE WITH ALL KNOWN HARDWARE/SOFTWARE

APOLOGY.

Some of you may have tried some of the small circuits which were published undocumented in the last two issues unfortunately I didn't get to check them prior to publication as Alan put them in as space fillers and I'm sorry to say the printer interface doesn't work and indeed isn't as far as I'm aware needed and the other which is a joystick interface does work but you can do away with 5 of the diodes and parallel the two Atari sockets together. I'm sorry if anyone has been misled we will try to do better in future.

HARDWARE SALES

I have decided to group all of my sales activities together under the banner of M.G.C.S (Mike Goddard Computer Support) to make things easier for myself and hopefully simplify things for members. I shall still carry on with all the usual activities for the group and hopefully provide a better service with more things being offered at a discount to members.

*Mike*

CASSETTE REVIEWS.....

NICKY GUDDARD

All of the games reviewed here are available from the group cassette library at the current library terms.

STAR RATING GUIDE.

One star = terrible, Two stars = "OK", Three stars = quite good, Four stars = very good, Five stars = Brilliant.....

ARTILLERIE

Library No G16

You control 2 guns opposite to each other positioned on grass which has dips and bumps which change at the beginning of each game. You fire your guns by stating the position of which you fire and what speed you want the bullet to fire at. Everything is in German but VU means speed and X means what position you want to fire at. The speed is in M.P.H. the maximum is 35 M.P.H. and the position of which you fire is in numbers also ranging from 1 which is straight in front of you to 89 where the guns fires the bullet upwards and it lands by the side of you. You win when you hit the opponents gun. When the game is over you press J to play again. A very good game, runs in Extended Basic but is slower than most joystick games.

STAR RATING \*\*\*\*

BATTLEFRONT

Library No G27

This quite a good game where you are a white rectangle shaped object and your aim is to destroy as many vehicles as possible. When you press the 'S' key you move straight across a stretch of land using 'S' or 'D' to move yourself from side to side. When you hit a vehicle you score some points and your bonus goes up. After that your bonus starts to decline until you hit another vehicle where you score in the same way. If you hit a building or miss the buildings and the vehicles you lose bonus points but your score stays the same. When there are no more vehicles to shoot the computer counts down from ten and then gives you another screen full of vehicles and you can start firing again but your score stays the same until you lose all ten lives and you have to start another game where the computer also counts down from ten. On the second screen of vehicles you get missiles fired at you. You can use joysticks in this game. A very good Extended Basic game which I hope to play a lot more in the future.

STAR RATING \*\*\*

BOGGLE CHALLENGE

Library No E2

I think BOGGLE CHALLENGE is quite a good name for this game because if you don't know how to play it it can get quite mind boggling. After you've selected how many players there are going to be (the maximum is 4) a 5X5 grid appears on the screen. In each square is a letter. At the top right hand corner of the screen is a clock counting down from 2 minutes to time up in 10 second steps. The object of the game is to make a word out of the 25 letters on the screen. If when the seconds are up and one player hasn't yet made a word he gets a penalty point. After the game when you've told the



computer who gets the penalty point it displays the players names and how many points they've got. The player with ten penalty points loses the game and the computer prints on the screen the players name and losses by the side e.g David loses which it repeats until you press enter when it starts all over again. It is a very good basic game which I enjoy playing  
 STAR RATING \*\*\*

THE CHESTER WORKSHOP..... Mike Goddard  
 =====

Well we finally did it !! held a hardware workshop/local meeting that is. It was held on the 12th August at Ken Wilfords house at Blacon in Chester, all of the members in easy reach of Chester were invited (if you were missed out I'm sorry it is difficult to judge who is close or not) by letter and all committee members notified. Some members who were invited but couldn't make it asked to be notified of any future meetings so it looks like we'll have to hold another!!

Subjects worked on and discussed were Joystick adaptors, keyboard diodes, reset switches and load interrupt buttons.

A good bit of time was spent discussing some of the mountain of literature I took along (fuel for future articles) of course the subject of expansion came up, but I had just recieved details of the Zenoboard and I think it will solve some of those problems (see article this issue).

As is often the case the afternoon went all too quickly and we probably didn't do all that we wanted to but we thought it a good idea to hold another in future but possibly in larger premises not that Kens house was too small quite the opposite in fact but if many more had come there would have been problems.

Finally thanks to Ken for being a very good host and providing excellent refreshments *Mike*

See article by Peter Walker pp.3,5, & 46  
 There are many types of conversion that you may need in your programs. If you want a few challenges, you might like to attempt the following:

1. Hex to Decimal convertor (and vice versa)
2. Radix 100 to Decimal convertor (and vice versa)
3. Roman numerals convertor (and vice versa)

and for disk owners:

4. Database convertors (eg PRK to TI-Writer)

In the next issue I will publish the best crafted and original conversion program submitted to me, be it one of the above or any other conversion idea.

RAMBLES

Written for Oct 89 TI\*MES by Stephen Shaw.

10 Alstone Road, STOCKPORT, Cheshire SK4 5AH

Welcome to another issue of RAMBLES.

Good to see so many articles flooding in for our magazine... together with a lot more activity by me, this means that while I have a great deal to tell you I have less and less room to do it! In the last two issues of TI\*MES for instance, only half the text I have sent in has been printed. Obviously I can't take up the whole magazine! but you will see the problems! If you have a disk system you can obtain RAMBLES on disk- in full!

The Sydney group have Rambles on disk, and therefore provided the first published source of an article I wrote for Rambles on Turbo Pasc 99 which appeared in our issue 25.

The Annual Show, back in June, is now ancient history, but you may be interested in knowing that we had some 20 unexpanded visitors and 45 expanded visitors- given a membership of about half unexpanded, and some remarkable deals on modules (good modules, in quantity) the lack of unexpanded owners was rather a pity.

Jim Peterson sent over a box of disks for us to sell- duly opened by Customs, who rather oddly, took one disk out of its envelope, tore it, and put it back into its envelope? I cant see how Jim could have recorded on such a badly damaged disk, nor how a disk can be damaged and leave no mark on its tyvek envelope...  
 Taxaments sent along over US\$400 worth of software for us to sell, but made the error of declaring a value of US\$100 on the Customs label. The package was opened for inspection and promptly seized.

MICROpendium intended to send us some sample copies, but forgot. Laura mentions that they have a mere 14 subscribers in the UK (the majority recruited by me!), which is a very low figure even considering OUR membership number. This is a good regularly published and well established magazine covering all interests including basic only, and with timely news of new products. I have been recommending this magazine for four years now- with apparently little effect.

Dolores of Harrison Software actually came along to be with us, all the way from Maryland (near Washington DC) and sold a few Music disks and a couple of word processors. Perhaps the members who purchased those would review them? Dolores has also submitted an article for TI\*MES on her music programs- watch out for it if and when space permits.

Bringing over a couple of boxes of disks for sale, she had to pay VAT at the airport on the retail value, and was promised a refund on any unsold, on exit. On leaving the UK, she got a non-English speaking Indian customs officer, who took her receipt and was never seen again. When you have a Jumbo waiting for you you are not in a position to hang around!

Dolores also suffered from other English bureaucratic bungling- while over here she wished to use our Public Record Office in London, and had a letter from them advising her that she could do so on production of her passport for id. When she got there they THEN insisted she produce a document from a University or Government Department vouching for the seriousness of her research (this is contrary to published information sold by the Government publisher HMSO). Does any member have access to the PRO please?



Lory also wanted to visit the British Museum reading room, but they chose the day of her visit to put on show a vase smashed up some 145 years ago, and she was unable to get near the museum! She did manage to visit J C Bach's grave.

Repeat- Lory would appreciate help with US/UK/German history around 1780/90, especially London, Hamburg and the Battle of Yorktown. Any lists of soldiers at the Battle? Any period music from any army, recorded or scored? Any access to Middlesex legal records of the period or details of legal practices especially costs? Activity of Catholics in the UK at this time, especially in London?

Why another word processor? Bruce Harrison describes TI Writer (quite accurately) as a TEXT EDITOR. As he was used to using a Wang Word Processor, he felt the need for something a little more like that - the entire document is NOT stored in memory but is page-oriented and saved and loaded from disk in page sized chunks. It took him two years to produce the present product, and as of June 89 some two dozen copies had been sold, two in the UK - review please chaps! - Bruce says that it has sold on the ease of initial use. With a half page ad in Micropendium at US\$124, this little family firm are a little uneasy about getting back their expenditure... their wordprocessor has an interesting EMBED command, which allows you to type in-just like this:

CHR\$(27)"A"CHR\$(8) or if you prefer just (27)"A"(8) and have the control characters for setting line spacing at 8 dots, placed in the document. They have also supplied conversion routines to and from TI Writer files, and can handle many standard Formatter commands- unlike TI Writer, their formatting is dealt with in normal printing rather than through a separate program.

I have been very busy still with Fractal graphics of all sorts, using Myarc XB, and if space permits there will be listings and book reviews. Fractal Report Issue 2 had an interesting article on how the data statements are derived which draw triangles, trees, ferns, scrolls and so on. When I have a spare week or so I just may be able to follow that and have a go on some fractals of my own...

Just about half way to putting a full index to TI\*MES into TI BASE format- up to issue 13! I have full indices (commercial) for 99er Magazine, Enthusiast 99, Super 99 Monthly and The Smart Programmer, and also have all the copies. I also have full (commercial) index to MICROpendium, and have all copies apart from the first year (1984). So if you need any information I may be able to find it for you!

Once I have completed indexing TI\*MES, I shall then add TI User and TI-Lines, both of which I hold complete. My collection of PARCO magazines ceases at Vol 2 No 2- are there any after this?

Also took time to look more closely at LISP99 (book review later if space) and it turns out to be a good implementation of MacLisp, just missing the Property List functions- also Quotient has the arguments reversed, and MINUS seems to be missed out! Otherwise you could say it was very like TI LOGO without the graphics!

Many thanks to Jan Alexanderson for sending along a seven page article on the Myarc disk controller, which I am sending along to the editor. Also thanks to both Ray Kazner and Ken Gilliland for sending along Kens three articles on the use of THE PRINTERS APPRENTICE, similarly forwarded to the editor with this text.

Our friends in Holland have sent along their Tijdingen, which is alas entirely in Dutch, which I do not speak. Anyone wish to borrow it to translate? I cannot comment on the articles, but the programs include a long Pascal (apparently TurboPasc99) program which has something to do with a calendar, lots of little routines for G (the graphics language from your disk library), TI Basic and ExBas programs for sound, 2D Cube, "hoesjes printen", and a program which appears to be nothing but Dutch Display Afs! There is also an article on Dutch Logo .

1st August sees the 1988 Patent Design and Copyright Act becoming law- this is 12.50 from your local HMSO. It has an interesting clause- not only is making of unauthorised copies unlawful, but in addition it is unlawful to sell or provide a product (which could include a software disk) which is DESIGNED to make an unauthorised Copy or which you have REASON to know that it will be used to make copies.

DM1000 and every other disk copier except the TI Disk Manager readily and easily ignore the simple copy protect features TI originally provided- however, TI provided an excellent track copier in their officially released TI FORTH, so mere track (and presumably sector) copying is probably lawful. The sellers of commercial copiers of protected programs (for other machines) are taking the view that they can continue to sell provided their product has a legitimate use and they do not advertise its unlawful possibilities.

Apologies to those of you who tried to telephone me in July- I was not on holiday - that was in August- but rather my telephone was u/s, courtesy BT. I don't get a lot of calls and apparently no one thought to report a line fault, so it may have been quite some time that you were getting an unanswered ringing tone- the phone just was not ringing.

MINI-REVIEW: PagePro from Asgard:  
This is a program which allows you to mix text and pictures on a page AND SEE THEM BOTH ON YOUR TV SCREEN together! There are a few limitations- the fonts and pics have their own format, but can be converted from TI Artist fonts and pics. The most significant limitation for me is that fonts MUST be either 8x12 or 16x24 pixels.

Why is this a limitation? The standard 40 column screen used by TI Writer is based on characters 6x8 pixels. The standard 80 column printer also uses a standard character definition of 6 pixels width, so using PagePro you become limited to a maximum of 60 character from page edge to page edge- not too bad for a flyer or advert or if you have little to say, but a bit of a loss when you wish to cram a lot of information into a small space!

Apart from that the program is remarkably easy to use. One other limitation- for best effect all characters should be centred in the 8x12 block, otherwise the printed effect can look- perfectly illustrated in the manual- like this:

```
LOADING INSTRUCTIONS.
No, there are no spaces after the I's, it's just that in TI Artist format they are firmly placed at the extreme left of the 8x12 block, leaving rather a lot of nothing after them! It's up to you to reposition the characters before you translate them to PagePro format!
```

PagePro has line wrap but not word wrap, perhaps making hyphenation a little easier! Import of DV80 text is possible WITH CARE!

Not an expensive program at around US\$25 or so, and certainly the fastest in its field if you wish to mix text and graphics.

MANY THANKS to those members who made their rigs available for the A G M.

Speaking to Larry Connor I gather that all the extra goodies for TurboPasc 99 -eg Windows disk and Utilities disk giving access to graphics and sound- are due for release at this years Chicago show, and therefore should be available by our NEXT issue. All registered purchasers of TP99 will be advised. The reason for the delays is the difficulty of translating technical material from German to English, and at the same time retain the meaning and improve the readability. These should make TP99 into a very worthwhile language indeed.



CORRECTION: Issue 25, Page 42: Triton XB example program.

```

Issue 25:
150 REM PRESS E TO EXIT
160 CALL COLORS(2,15)
160 PRINT "PRESS S TO PRINT SMALL PIC"
should really read:
150 REM PRESS E TO EXIT
155 CALL LINK("SHOW")
160 CALL COLORS(2,15)
165 PRINT "PRESS S TO PRINT SMALL PIC"    Sorry.
    
```

**THERMAL TRANSFER RIBBONS** or how to make your own T-Shirt Graphics: Some time ago, Personal Computer World Magazine gave the address of a California company who sold special printer ribbons which you could use to make your own transfers for your very own T-Shirts. In due course, TI supplier TEX COMP also made these available, but only in one package, whereby you had to buy a black ribbon with a set of coloring pens.

I have now located a UK source of these ribbons! For the Epson MX80/FX80 series the ribbons are eight pounds each including vat but exclusive of postage. They are fabric ribbons, but to produce a good image usage is limited to at most 100 images, and possibly as few as 20, depending on size and density.

BLACK ribbons are available as one-offs, but there is a minimum order of four-off for each of the coloured ribbons, which are inked to order. Colours available are red, brown, blue, green and purple.

To use, you print in the usual way onto a heavy smooth paper (70/100 gsm). The cloth should be 50/50 poly-cotton: pure cotton is unsuitable, more polyester may be better but causes difficulties when transferring the image as it scorches more easily.

Place foil on the ironing board, then your cloth, then the paper with the image, face down, then two sheets of plain paper to spread the heat. With your iron heated to "COTTON", heat the image for 30/40 seconds and the transfer is complete. Maximum image size for a normal domestic iron will be about 6"x6". They also supply the heat transfer PEN set (you colour the PAPER you have printed on and then transfer the coloured image), and ribbons for many printers such as LX80, MX100, Citizen 120D, Panasonic 1080, Star NX10 etc.

Due to the minimum order for coloured ribbons I have only tried the black ribbon so far- this gives an initially brownish/black image but the brown washes out leaving a black image. Wash is "COOL" and ironing is not advisable but if required should be at lowest temperature and iron on back of image not face.

I have not had my ribbon long enough to comment on long term durability of the image, but my test piece has gone through one test wash without difficulty.

If members would like patches with words or pictures on to sew onto their clothing, baggage etc, I have some material I can impress the images on and send on to you. Just send along your image (any format but TI Artist is best!) and indicate printed size where the format allows choice of printout size!). To cover cost of ribbon, material and electric! please send a minimum donation of 60p, more if the image is largish or fairly dense. Larger donations welcome. DO NOT send clothing- I don't wish to scorch your valued Tibetan yak-hair smock!

You can always buy your own ribbons! (-->continued-->)

From the same supplier, further unusual items- if you have a printer which insists on tractor feed only, you may feel you cannot use lovely coloured or printed papers- not so. The choice of "continuous" paper is as follows:

Parchment in Lavendar, Natural and Turquoise (also packs available with seals and ribbon), Pastel colours-vanilla, orchid, pale blue, pale yellow, green; bright colours: yellow, red, blue, green, pink; also aqua, buff, ice blue, ivory and pearl gray to match printed papers below.

And printed paper (with pictures printed on) as bears going camping, bears holding hearts, bears with bow ties, butterflies, christmas geese, christmas morning, dinosaurs wearing sneakers, enchanted flower garden, holly border, jolly snowmen, lambs playing, seagulls soaring, biplane, lilies, mountainscape, ship, tropical beach.

Matching envelopes available of course! Prices vary a little- for the printed papers you can buy as few as 50 for 8.95 or 250 for 16.95. 250 Pastel shades are 10.95.

Oh yes, the supplier... MGA MICROSYSTEMS, Pear Tree, Appledore, Kent, TN26 2AR. To discuss your requirements telephone 0233 83571, but they do not take credit cards!

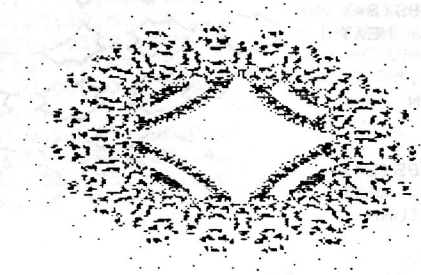
**FRACTAL IMAGING PROGRAMS**

There are a number of ways of obtaining bit map graphic images on the TI99/4A, and while I have used Myarc XB for these programs, they should be readily adoptable to any other form of program that allows the use of the 192 x 256 pixel graphics screen, such as Triton XB, or JBM103 (available from the disk library, and requiring only TI XB and 32k), or the Australian graphics language G, or perhaps even a Geneve!

The first program is a variation on a program I submitted in several formats for the last issue. This one uses two very slightly different forms of the basic function, and sets the variables at random. Scaling to the screen is automated and is based on the first two hundred points plotted.

The second program produces some interesting images, far more complex than the code may imply- a tree, a fern, a Serpinski triangle, and a black square. The square is of interest as depending on how many iterations you allow, that sets the density of the square (try it). Kicking in the random probability subroutine will demonstrate how to turn the black solid square into a set of patterns- one particular set of probabilities will give you a straight Serpinski Triangle! The routines for Q=5 and Q=6 are also squares, but the images are derived with (5) different base data and (6) preset different probabilities.

Instead of allowing random operation you can of course input data as you see fit. The CALL LINK("DUMP") is a machine code routine for Myarc XB which is available from the disk library which allows you to dump a Myarc XB bit map screen to an Epson printer via PIO.









```

710 DX=X*2+40 :: DY=Y*1.92 :
: RETURN
720 DX=X*1.3+40 :: DY=Y*1.3+
50 :: RETURN
730 DX=X*40+120 :: DY=193-Y*
19+5 :: RETURN
740 DX=120-X*250 :: DY=200-Y
*380 :: RETURN
750 DX=X*150+60 :: DY=Y*150+
30 :: RETURN
760 DX=190-X*98 :: DY=80-75*
Y :: RETURN
770 DX=X*160+30 :: DY=Y*160+
25 :: RETURN
780 ON ERROR 780 :: RETURN N
EXT
790 END
    
```

**CORRECTION!!!!**

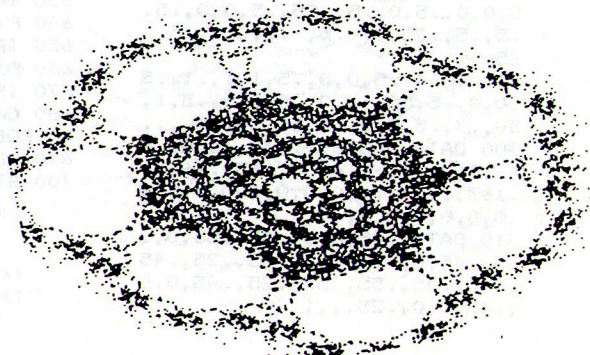
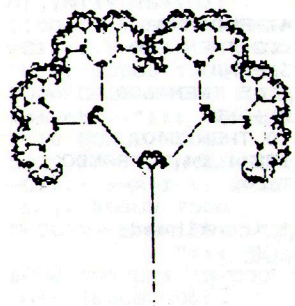
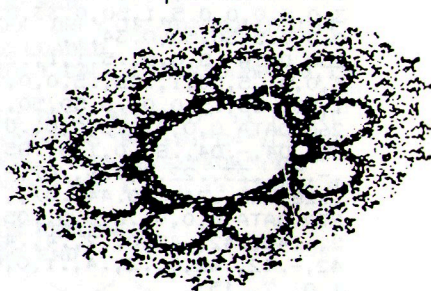
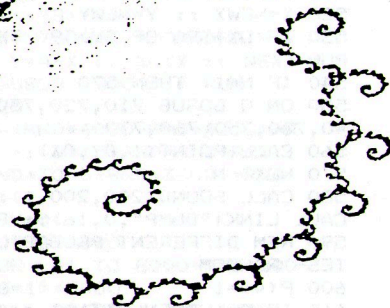
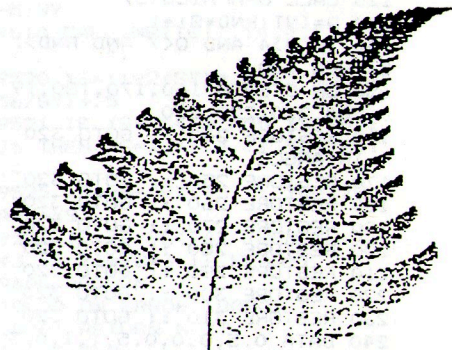
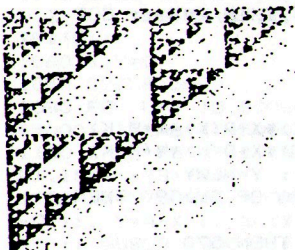
In the above listing I seem to have lost a line...

PLEASE ADD:

```

440 IF K<P(1) THEN K=1 :: GOT
0 480
    
```

Sorry bout that- at least I spotted it just before submission this time!



RAMBLES TEST 1 -

Here is the explanation of the fastest Extended Basic routine given in the last issue, which was:

```

100 FOR N=32 TO 99
110 IF N*(N-1)=99*INT(N*N/100) THEN PRINT N*N
120 NEXT N
    
```

Thanks to John Stocks for the explanation of the code, which originated with Peter Walker:

The numbers we are looking for are (by definition) perfect squares, so if we consider  $N^2$  ( $=N*N = N$  squared) to be the four digit number ABCD, with AB being the number of hundreds:

$$AB = \text{INT}(N*N/100)$$

$$CD \text{ is } ABCD \text{ less the number of hundreds:}$$

$$(CD = N*N - 100*AB) \text{ hence:}$$

$$(CD = N*N - 100* \text{INT}(N*N/100))$$

Adding  $AB+CD=ABCD$  as defined above. Adding the two formulae:  
 $AB+CD = \text{INT}(N*N/100) + N*N - 100*\text{INT}(N*N/100)$

This reduces to:  
 $AB+CD = N*N - 99*\text{INT}(N*N/100)$

We are looking for  $ABCD=(AB+CD)^2$

$$\text{i.e. } N^2=(AB+CD)^2$$

$$\text{i.e. } N = (AB+CD)$$

$$\text{i.e. } N = N*N - 99*\text{INT}(N*N/100)$$

$$\text{i.e. } 99*\text{INT}(N*N/100) = N*N - N$$

$$\text{i.e. } 99*\text{INT}(N*N/100) = N*(N-1)$$

HMMM. It does speed things up though! Only one further submission up to this press date, so here are all the other language versions received to date:

**FORTH-** John Seager:

```

: ANSWER CLS 0 0 GOTOXY ." WORKING" CR CR
10000 1000 DO I DUP 100 /MOD + DUP * OVER
= IF . CR CR ." WORKING" CR CR ELSE DROP
ENDIF LOOP ." DONE" ;
    
```

```

C99 - John Seager
/* requires csup and printf */
extern printf();
main()
{
    
```

**PASCAL-** Turbopasc99 - John Stocks:

```

PROGRAM testpas;
VAR n:INTEGER;
    m:REAL(4);
BEGIN
FOR n:=31 TO 99 DO
BEGIN
m:=n;
IF m*m-99*INT(m*m/100)=m THEN
writeln(n*n);
END
END.
    
```

```

int i,a,b;
printf("working\n\n");
for (i=1000;i<=9999;i++)
{
a=i/100;
b=i%100;
a=(a+b)*(a+b);
if (i==a)
{
printf("%d\n\n",i);
printf("Working\n\n");
}
}
    
```

(This can be speeded up a mite)



MACHINE CODE- SOURCE: John Stocks:

```

DEF TEST
REF VSBW,KSCAN
TEST LI 0,>00EC
LI 4,>0044
LI 10,>001E
A INC 10
CI 10,100
JEQ C
MOV 10,11
MPY 10,11
MOV 12,13
LI 14,>0064
DIV 14,11
DEC 14
    
```

```

MPY 14,11
NEG 12
A 13,12
C 12,10
JNE A
MOV 13,3
A 4,0
LI 5,4
B LI 1,10
CLR 2
DIV 1,2
MOV 3,1
AI 1,>30
SWPB 1
BLWP @VSBW
    
```

```

DEC 0
DEC 5
MOV 2,3
CI 5,0
JNE B
JMP A
C LI 1,>2000
D BLWP @KSCAN
MOV @>837C,2
COC 1,2
JNE D
CLR 0
MOV 0,>837C
LWPI >83E0
B @>70
END
    
```

If you can find any routines in any language faster than the routines for that language published here, or in any additional language, please send them in to Stephen Shaw.

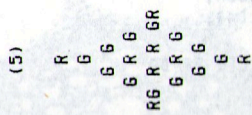
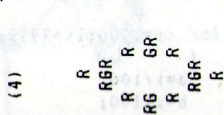
RAMBLES TEST2.

This is a real test, a program requested by an overseas member! I've had a bash at it, and the result (in Basic) is SLOOOOW. Let's see what you can do!

The requested program is to use a 32x24 grid of graphics characters, which may be red, green, or screen colour, in accordance with the following rules:

1. Initial layout of red characters is input by user, preferably by moving a cursor around.
2. User indicates how many generations (=loops) he wishes before the program terminates.
3. IF the program is fast enough! some means to pause and/or slow down.
4. Rules for pattern generation:
  - i. Check all no colour squares and if ONE side and one SIDE only is adjacent to a red or green square, then in Step iv below, and ONLY in step iv! this square is to become a new red square.
  - ii. change all green squares to no colour squares
  - iii. change all red squares to green squares.
  - iv. change all squares memorised from step i above into red squares.
  - v. go back to step i.

Easy huh! Lets see the fastest you can throw at us! So you can check the validity of your program before submission, here are the 4th and 5th patterns, assuming that the starting pattern, pattern one, is a single red square in screen centre:



Anyone working in machine code is requested to consider the 48x64 multicolor code screen, in addition to the 24x32 graphics screen.

DISK LIBRARY ADDITIONS

An up to date library catalogue is available on three disks- just send three disks and return postage for a copy. And don't forget to take a fresh look at some of the old items from time to time, as they may now be suited to your present needs and interests.

A print out of new disks received is mailed out on request every two to three months to any member who has sent along one or more SAEs for this purpose.

Library costs are:

Copying: One pound per disk side.

Disk if supplied by library: one pound per disk.

Postage: One pound per order (UK only)

eg Order five disks and send blanks with your order, cost is 5+1= six pounds.

Library address:

10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH.

Overseas enquiries welcome with two IRC's or US\$2 in notes.

>REBEL (TWO DISKS)- Lots of utilities and source code- Cassette Backup- copy a whole tape from CS1 to CS2-essentially an audio copy via the consoles audio circuits. Cassette Builder-uses a DV80 control file to copy several disk files onto one cassette automatically. CRU TESTER- both educational and diagnostic. DNAME-a m/c utility to incorporate into your own m/c programs, allows input of a disk name, checks validity and calculates length. Disk Buffer-allows LONG TI Basic programs to transfer from tape to disk, even when CALL FILES(!) fails to work. QUICKSAVE will change a DF80 file WRITTEN FOR XB into what looks like an XB program. SMALL LIST (also to be found on UTIL-21),allows you to list XB programs in 28 column format. SPEECH looks through the speech synth and speaks every word it can find.

>System Disk Loader (TWO DISKS) by EP Rebel, includes source code. Produces a custom menu as UTIL1 and has loaders for XB and MM. Loads only memory image machine code programs. No options so some it loads, some it doesnt- Scrabble is OK but Superfly is not. An alternative to Joe Nollans disk. Also a suite of CALL LINKS for XB programmers, to play with the screen display-the links are BIRMIR, BITREV, BITTRN, DELETE, DEMO, DOWN, DUMMY, HELP, INSERT, INVERT, LEFT, MIRROR, RANDOM, RIGHT, ROTATE, ROTNEG, ROTPOS, SCROLL, SORT, SWAP, SWPMIR, SWPTRN, TURN and UP, which allow you to do pretty well everything with a screen display! Up down left and right move the whole screen image and also wrap around.

>LOGOUTIL. TWO DISKS. The same utilities as are on UTIL-22 but here with docs and source code as well. Dump a graphic representation of LOGO Characters and Tiles. Make a Logo2 program autostart- no more fumbling for the name required to start a Logo program after you load it! Additional program REFDEF which lists the REF and DEF words for DF80 files, but will not function with Funlweb, as it checks which module loaded it, and does not allow for Funlweb.

>STATIONFALL the Infocom Adventure (DSSD only) now has added a very important cheat file- the Astrognational Chart, which you must have to chart a true course. Obtained from the release for Amiga- and the supplier would like to see you in the company of Witness, Suspect or Sea Stalker- we can



return the help, just let me know! Ta.

>COLLINS- taken from the Collins starter and games packs- Airship, Bat, Cards, Commando, Crosses, Dicerace, Dragon, Duel, Logical, Maze, Rabbit, Racetrack, and Target.

>SAMS ENTERTAINMENT GAMES- from the Sams book- Bio, CosmicGuns, Dungeon, Goldbag, Gunner, Homebound, Killcrab, Phoenix, S/A/M, Skeetshoot, Spritedef, Typing and Arrow Zap.

>SAMS. From various SAMS books-Bach, Blackjack, Boxes, Capitals, Charcode, Chargen, Comptutor1, Cubits, Dicegames, Dixie, Firstsong, Frenchfood, Geometry, History, Intro, Jellybeans, Lullaby, Metriccon, Patterns, Piano, Pin\_Ball(Lizard S/W), Rocket, Samsmusic, Scrambler, Simonsays, Simpledice, Sketch, Sounddev.

>B A WRITER Version 1.4 (TWO DISKS)- an alternative to Funlweb, this is from Paolo Bagnaresi of Italy. It is an amended XB loading version of TI WRITER. Myarc compatible.

>DISNEY01. TWO DISKS. FOR MILLER GRAM CRACKER ONLY. ET, ET AT SEA, FANTASY, PINOCCHIO, VONDRAKE.

>DISNEY02. TWO DISKS. FOR MILLERS GRAM KRACKER ONLY. GERM PATROL, MOUSEATTACK, PETER PAN( not in GK format- not sure what will load this, I dont have a GK!), Plant Genetics, and Wing War.

NB: As I do not own a GK I cannot vouch for the above two disks!!!

>TI WRITER Vn 4.3- modified Editor AND Formatter, with new formatter commands and much faster editor move, copy, delete and replace string- for which I shall be using this version while retaining Funlweb for main writing and environment. Includes special menu-retaining loaders for mini mem and super space. Will load from Funlweb but crashes on exit.

>GPL MANUAL- FOUR DISKS- The official TI info on GPL.

>RAGMAC- Macro Assembler is now upgraded to Version 7. THREE DISKS - Completely new documentation, an installation program, output to disk now permitted, two new directives, 8 new macros.

>RAG CASSETTE LOADER- programs to save to tape and load from tape memory image machine code programs- 32k ram required- , with options for suppressing the normal double recording (halving time) and compressing data- both incompatible with usual format.

>RAG LINKER is upgraded to Version 3. New installation program and output may now be to disk.

From Mike Curtis-

>SHAMUS MAPS- Four levles of mapping showing where extra lives and the keys to which doors are- requires Picasso to view/print. 17 July- XB print program added- slow! For Epson printer etc.

>DM1000: We now have back in stock Version 3.5, which is the last SAFE version of DM1000. There will be no reliable versions with higher version numbers. If there is a subsequent release it will be called DM2000, but news on that is not too good.

>USER GROUP LIST FOR TI BASE by Andi Wise, REQUIRES TI BASE database program. Not merely a listing of user groups but also a good demo of TI Base with many command files to peruse.

>NUTS AND BOLTS DEMO DISK. Auto load auto repeat demo of SOME of the routines from the Nuts and Bolts disks.

COMMERCIAL CORNER: These disks are copyright and are copied with ther consent of the copyright owner who will be paid a royalty. They are not to be copied by group members nor by other users groups unless consent is first sought and obtained from the author.

NUTS AND BOLTS 1.

NUTS AND BOLTS 2.

NUTS AND BOLTS 3.

Each package consists of TWO disks, one of which has documentation. Each program disk has about 100 ExBas routines in Merge format. Each package costs six pounds, including the costs of the disks.

TIPS FROM THE TIGERCUB 1

TIPS FROM THE TIGERCUB 2

TIPS FROM THE TIGERCUB 3

TIPS FROM THE TIGERCUB 4

TIPS FROM THE TIGERCUB 5

Each disk contains programs in ready to run form, and possibly a few text items, taken from the long running popular Tips from the Tigercub series, with the emphasis on programming tips in Basic and ExBas. Each single disk costs FOUR POUNDS including the cost of the disk.

I have one copy left (ONE!): TI BASE Vn 2.01 WITH MANUAL. 15.00 inclusive.

Normal copying fees again!:

>TI BASE TUTORIALS BY SMOLEY. (FOUR DISKS). These disks contain nine tutorial articles with supporting databases and command files ready to use on disk. Requires TI Base to use. Covers articles from Nov 88 to April 89, both versions 1 and 2 of TI Base.

>TI BASE TUTORIALS by GASKILL. (ONE DISK) Four tutorials on version one of TI Base, with a filler artical by Bill Gaskill on general TI things.

>THE GOLDBERG VARIATIONS (TWO DISKS) More superb music from Harrison Software, copyright reserved, copied with consent. Members and other groups asked not to copy unless consent is granted. About 90 minutes of fine music by J.S. Bach.

>RAG UTILITIES Vn 6 (docs refer to Vn 5!). Quite a collection of utilities by Art Green to: i). Produce a formatted listing of an XB program with cross reference of variables and statement numbers. ii).A sector based disk copy routine for SS or DS SD drives. iii)Disk initialiser for SS or DS SD disks. iv).Program to copy all PROGRAM type files on a disk to cassette without operator intervention. v).Printer initialisation routine. vi).Dumb ASCII terminal emulato at 300 or 1200 baud with 590 line buffer. vii).disk catalogue routine for up to 255 disks, listing by disk or sorted file. Show directory to show or print single directory. Shorten GK file- reduce files created by GK from a wasteful 34 sectors to the 26 sectors of info they contain. Print file will print any DV file.



>Added to GAMES 20 (now 203 sectors used!) is CRIBBAGE from Canadian J Frank.

>TIMELINE has been augmented and updated with an extra 59 sectors of data, with just 91 sectors left free. Why am I telling you this? Nobody so far (nobody!) has asked for a copy of this potted TI history...

>LISP99 is unaltered but I can now reveal that it is an implementation of MacLisp, less the Property List commands. Also each definition or command is limited to a standard TI input line (5 lines). Rather like TI Logo without the graphics!

MF ART SERIES- Instances for TI Artist, converted from the MacFlix series of disks over many hours by Neil Wilson- TI Artist required- largest instance is 12 sectors:

- >MFART1..42 files.
- >MFART2..43 files.
- >MFART3..43 files.
- >MFART4..43 files.
- >MFART5..43 files.
- >MFART6..41 files.
- >MFART7..14 files: 113 sectors used.

I could list all the file names but they are not all that meaningful! And distributing a print out will all the instances illustrated would be a trifle costly! The price per instance is low enough to take a chance eh!

>FUNLWEB 4.13 is still Version 4.13, but is now debugged up to 4th July 1989-- Load, Util and ED/EE are amended- allows use of RS232/3 and RS232/4 is fitted, and also corrects a minor bug when printing disk directory from SD. Two disks remember!

>TONY\_MCG1 - out goes 199 sectors of articles (now on Tony\_MG3) to make room for a revised 80 column ED/EE set of files, PLUS NEW QDAV- in addition to 80 column TI Writer,SD is now also 80 columns- but the files ONLY work on a PEB with Mechatronic or Diji 80 column card or Geneve computer.

>TONY\_MCG3 now has the article displaced from the above disk and as a result is now 261 sectors used.

>TEXTLOADER+EASLOAD- Textloader is now upgraded to Version 1.2, January 1989. Principal changes are better handling of text lines of exactly 80 characters, and ability to handle hard disk path names. Note that Textloader can load and run programs or commands from DV80 files without modification- no need to remove rems or spaces etc.

>ENHANCED DISPLAY PACKAGE is now up to Version 2.4, but as this is at present only in the US version (clock speed out for non-US use) please would you specifically request it as EDP 2.4US, otherwise the UK version 2.1 will be supplied. The author has not kept a record of the changes which appear more to be debugs than enhancements. Apart from the displayed clock running slow, this version is fully operational here. There MAY be a UK 2.4, but no guarantees!

NOTE: TI SORT advertised by Texaments for TI Base is described in the same way as, and times out identically to library disk SORT1. We are licenced to copy SORT1 (which is supplied with SPELL1 too!) which remains copyright and should not be copied by our members or other groups without the authors consent. The main difference between the library disk and the Texaments

disk: Our has SPELL1 on it as well; we have the original (sparse) documentation- Texaments probably have fuller docs, specifically related to TI Base; Texaments supply a set of loaders, our disk is only in EAS format- loadable by Funlweb. Texaments were advised of this sort routine by ourselves and made their own arrangements with the author.

Thats about it for now. Dont forget to read through the whole catalogue from time to time- your requirements NOW may be different to the last time you looked! An old commercial program- SORT1, which hardly sold at all and was almost unheard of, has now become an important utility for owners of the most popular TI Database! Some of these old programs are worth looking at!

Best wishes. Stephen Shaw

BOOK REVIEWS.

Note on purchasing: All the books below can be ordered from ANY good book shop, including most branches of W H Smith, and all branches of their subsidiary Sherratt and Hughes (and Waterstones) as well as Dillons, Hatchards, Willshaw and independants. You will probably need to leave a deposit- the final price will depend on the price on delivery to the bookshop which is usually within three weeks of order. It may be worth ringing (free) The Book Company on 0800 010910 who supply many books post free by mail order.

THE ARMCHAIR UNIVERSE. A K DEWDNEY. W H Freeman and Co. Paperback. 330 pages incl index. ISBN 0-7167-1939-8. About Pds 11.00

This book has been cited in Rambles twice already- in association with Wa-Tor and with Fractals, so I thought it was time I bought a copy! It is based on a series of computer-related articles in Scientific American, and is full of interesting ideas on things to do with your computer. Section headings include Infinite Graphics, Mathemagadgets, Artificial Intelligence and Artificial Insanity, Life in Automata (imagine a game of 3d animated life...), Puzzles and Wordplay, Stimulation through Simulation, and Core Wars.

A good read for the computer literate, it also gives some ideas for programs to play with- with the emphasis on recreational programs rather than games or utilities. Very interesting reading.

LISP-A GENTLE INTRODUCTION TO SYMBOLIC COMPUTATION. by David S Touretzky, Published by Harper and Row, 384 pages incl index. Paperback. About Pds 15.00

This is the book recommended for LISP99, and is based on MacLisp, as is LISP99 (available from your user group library!). An excellent introduction both to Lisp and to Lisp 99, easy to follow, apart from an awkward Chapter 0, and maybe a different approach to Chapters 1 and 2. Start with Chapter 3 with Lisp 99 loaded and ready! Only a chapter towards the end- Chapter 11 on Property Lists- is of no value with Lisp99.

Lots of exercises, WITH ANSWERS! All inputs to LISP99 are in ROUND BRACKETS, as shown in the book eg:  
(DEFUN DOUBLE (Y) (MULT 2)) defines function DOUBLE  
(DOUBLE 5) gives the output then of 10.

FRACTALS EVERYWHERE by Michael Barnsley. Hardback. 393 pages incl index. Academic Press Inc. ISBN 0-12-079062-9. About Pds.27.00

OK, this IS an expensive book, no doubt of that, but if your imagination is



grabbed by fractal graphics, it is possibly the best value book you can buy, as it has lots of algorithms and IBM Basic listings, easily converted to Myarc XB, lots of sample graphics in black and white AND colour, and an appalling amount of pretty heavy going math, BUT presented in a pleasant chatty style which is not at all frightening! Lots of exercises but NO answers! My printer won't go anywhere near duplicating some of the math symbols used, whose names I do not know, but understanding the math can be left until after you have thoroughly explored the listings, there is so much to consider that even if only a little of the theory sinks in, you have greatly advanced your logical or mathematic thinking! Dont worry about things like "The random iteration algorithm is founded in ergodic theory"- it paints a pretty picture!

THE SCIENCE OF FRACTAL IMAGES. Many authors! Pub. Springer-Verlag. Hardback. 312 pages incl index. About PDS 23.00 This one has a foreword and a chapter by Mandelbrot himself, but also chapters by others extending well beyond the famous Mandelbrot set. This book is rather based upon the idea of looking at something and then determining a fractal approach to imaging it - not such light reading! but some very interesting colour plates. There are lots of algorithms presented as well, which appear to be in Pascal or a similar language. The math is much heavier than in Barnsley's book, with section headings such as "The spectral representation of random functions". Some interesting information in here though- watch out for fractal images in STAR TREK II next time you see it- watch for TINY triangles forming the image! Escher, the graphics artist, made use of fractals for some of his designs.... there is also a chapter on having an eye for fractals when taking photographs. This is a book for the more dedicated fractalist.

TI BASE BOOKS...

Well, not quite... but TI Base is a CLOSE approximation of the old database language called dBase II, and although most books are now on dBase IV plus, there remain a couple of books on dBase II around still.

At present TI Base lags behind dBase II in not having a report-form capability (understood to be in progress) and only limited screen set up commands- no form creating commands such as PICTURE.

The CHEAPEST book is:

WORKING WITH dBASE II. M de Pace. BSP Professional Books. Paperback. 175 pages incl index. ISBN 0-632-02305-8. About Pds 10. Despite the differences between dBase II and TI Base, this book gives a useful introduction to creating and using databases in this type of environment. It is interesting to see that TI Base has more math operators, but fewer functions.

UNDERSTANDING dBase II. Alan Simpson. Sybex. 260 pages incl index. Paperback. ISBN 0-89588-147-0. About Pds 22. Quite a bit costlier, but I think its rather a better read, with clearer lay out and progression. This is my preferred book.

Incidentally- the TI 99/4A uses DSK1 where the rest of the world uses B:, so if you see USE B:MAIL, on the TI you would type USE DSK1.MAIL

The disk library has a four disk set of TI Base tutorials by Smoley, with sample databases and command files, which is more relevant than any book, and a lot cheaper!

There is a good Pascal book by Doug Cooper called OH PASCAL, which is appropriate to Pascal 99- I already have a good Pascal book so did not buy

it, but I did buy a companion book- TEACHING INTRODUCTORY PROGRAMMING (with Oh Pascal 2nd Edition). Doug Cooper.

W W Norton and Co. ISBN 0-393-95447-1. Paperback. 173 pages. About Pds 5.00

-yes a computer book for a fiver!

This book is intended for teachers using Oh Pascal as a text book, and while the chapter on Staff Meetings is utterly useless, what gives this book value is a series of Pascal test papers and homework assignments- about 107 pages of tests of one sort or another. There are NO answers but if you work your way through this lot, you should end up with a fairly solid idea of what Pascal is all about!

COMPUTER SCIENCE LOGO STYLE. Three volumes. Brian Harvey. MIT PRESS. Paperback.

Volume One: Intermediate Programming. ISBN 0-262-58072-1. 315 pages. About Pds 16.00

The chapter on Logo versions lists no less than 18 versions, and the TI Logo gets no mention at all, which gives you an idea of how recent the book series is! Consequently, as the other Logo's have evolved away from the older TI Logo, some of the examples will not run without some alteration, if at all possible.

The series is an interesting up to date text on Logo though, well set out and eminently readable. Where there are major advances in Logo, perhaps the series may inspire a TI programmer to write a more up to date TI Logo in machine code?

G

The Graphics language from Adelaide Available from the User Group Library..

These simple routines are taken from TIjdingen July 1989.

The headings are in Dutch, which I dont understand, but rather than omit credit to anyone I will include them in full before the routine:

Apart from routines using ARC which is not generally available, you should be able to transfer these images to other languages making bit map or pseudo-bit map graphics available such as Myarc XB, Triton XB, TI XB+JBM103, Logo etc etc.

```
1. 3D BDL : Bron MSX Computer Magazine 3e jaargang nr.16
Auteur:H.J.Smit,Amsterdam. Vertaald:Ronald van Kleunen Buys Ballotstraat
27, 4462 AM Goes. "Regio Zeeland"
FOR N=5 TO 70 STEP 5
ARC 190-N 100 N N 0 360
ARC 50+N 100 N N 0 360
NEXT N
STOP
```

```
2. 3D-MSX : Heading as No 1 above.
:START
COLOR 7
FOR N=0 TO 7
A=N*5 ARC 128 85+A 85-A 85-A 0 360
NEXT N
```

CONTINUED->



```
FOR I=1 TO 600 NEXT I
CLS
COLOR 8
FOR N=0 TO 21
A=N*5 B=45-N*2 ARC 128 125-A B B 0 360
NEXT N
```

```
FOR I=1 TO 600 NEXT I
CLS GOTO :START
```

3. ARTISTiek: Bron: KIJK Nr 10 okt 1987. Auteur: Gilbert Stokman  
Orgineel: MSX Rest of heading as Nol.

```
FOR X=0 TO 250 STEP 5
A=X/1 B=X+A SET X X TO B 180-X
SET 256-X X TO 256-B 180-X
C=X/2 SET 125 X TO 125-C 192
SET 125 X TO 125+C 192
NEXT X
STOP
```

4. ATARI grafisch Bron KIJK No 8 aug 1986 Auteur Arvid Riem Lelystad  
Orgineel Atari 800XL (Kleuren verwijderd). Rest of heading as Nol.

```
B=0
FOR L=1 TO 79
SET L+90 B TO 160-L B
SET L+90 B TO 160-L 190-B
SET L+90 190-B TO 160-L 190-B SET L+90 190-B TO 160-L B
B=B+6 IF B>190 THEN B=0
NEXT L
STOP
```

5. ESCHER Driehoek van M C Escher Bron KIJK Nr 10 Oct 1988  
Auteur Roy van Ginkel uit Zutphen Rest of header as Nr1.

```
SCREEN 5 COLOR 1
SET 75 139 TO 92 150 TO 185 97 TO 185 72 TO 92 21 TO 75 30 TO 75 139
SET 92 150 TO 92 62 TO 132 87 TO 110 100 TO 110 119 TO 185 72
SET 132 87 TO 151 76 TO 151 76 TO 75 30
SET 110 100 TO 110 74
STOP
```

6. DEMO1 Jan Knapen

```
LINestyle 7 ARC 100 100 75 75 0 361
LINestyle 5 ARC 100 100 50 90 140 501
LINestyle 2 ARC 100 100 90 50 440 801
LINestyle 4 ARC 100 100 60 60 0 361
```

```
FOR I=1 TO 1000 NEXT I
CLS
SCREEN 15 FGND 4 BGND 9
```

```
:NE
REM :NE MUST BEGIN AT FAR LEFT MARGIN!
FOR I=0 TO 4
TRACE I SET 0 20*I+25 TO 255 20*I+25
SET 20*I+25 0 TO 20*I+25 191
NEXT I
GOTO :NE
```

7. DEMO2 Jan Knapen

```
SET 36 4 FOR I=0 TO 270 STEP 90
ANGLE I DRAW 184 NEXT I
ANGLE 0 SET 36 50 DRAW 184
```

CONTINUED →

```
SET 36 96 DRAW 184 SET 36 142 DRAW 184
ANGLE 90 SET 82 4 DRAW 184
SET 128 4 DRAW 184 SET 174 4 DRAW 184
FOR I=1 TO 5 PRINT I 8*I "POS" NEXT I
FOR I=15 TO 20 PRINT I 8*I "NEG" NEXT I
BOX 1 4 30 184 PATTERN 8440 FILL 9 12
CLEAR 2 5 28 182
PATTERN 8320 FILL 40 8 PATTERN 8320 FILL 88 10
PATTERN 8320 FILL 136 12 PATTERN 8320 FILL 184 14
PATTERN 8232 FILL 40 54 PATTERN 8240 FILL 86 54
PATTERN 8248 FILL 132 54 PATTERN 8256 FILL 178 54
PATTERN 8264 FILL 40 100 PATTERN 8272 FILL 86 100
PATTERN 8280 FILL 132 100 PATTERN 8288 FILL 178 100
PATTERN 8296 FILL 41 147 PATTERN 8304 FILL 86 146
PATTERN 8312 FILL 132 146 PATTERN 8320 FILL 178 146
INVERT 24 24 155 155
STOP
```

These demo routines also illustrate and amplify the rather limited documentation supplied with the G package!

#### LOGO

LOGO ROUTINES : with apologies to the original author whose name has become detached from the routines (sorry):

```
TO DRAWIT
TELL TURTLE HT CS CB 10 SC 13
SXY -42 -42 PD
LT 30 REPEAT 3 [LLL RT 60 ]
END
```

TO LLL

```
FD 15 RT 60 FD 100 RT 120 FD 85
RT 120 FD 15 RT 60 FD 55 LT 120 FD 85 RT 120 FD 15
RT 60 FD 100
END
```

=====

----second logo routine----

TO TSTAR

```
TELL TURTLE PD CB 4 SC 5 CS HT SXY -45 -47
REPEAT 3 [BLOCK1 PU LT 30 BK 14 LT 90 FD 7 PD ]
PU FD 22 RT 120 BK 15 PD
REPEAT 3 [BLOCK2 PU LT 30 BK 18 LT 90 FD 6 PD ]
PU FD 18 RT 120 BK 12 PD
REPEAT 3 [BLOCK3 PU LT 30 BK 16 LT 90 FD 9 PD ]
PAUSE 150 REPEAT 11 [STAR PAUSE 10 PR ]
END
```

TO STAR

```
SXY -26 9 SH 0 FD 12 LT 60 FD 10 RT 120
FD 9 LT 60 FD 10 RT 120 FD 11 LT 60 FD 10 RT 120
FD 11 LT 60 FD 9 RT 120 FD 10 LT 60 FD 11 RT 120
FD 10 LT 61 FD 11
END
```

TO BLOCK1

```
REPEAT 2 [FD 21 LT 60 FD 16 LT 120 ]
```

CONTINUED →

```
RT 60 FD 16 LT 60 FD 21 LT 120 FD 16
RT 60 FD 16 RT 120 FD 16 RT 60 FD 16
END
```

```
TO BLOCK2
REPEAT 2 [FD 16 LT 60 FD 16 LT 120 ]
RT 60 FD 22 LT 60 FD 16 LT 120 FD 22 RT 60
FD 16 RT 120 FD 22 RT 60 FD 16
END
```

```
TO BLOCK3
REPEAT 2 [FD 16 LT 60 FD 22 LT 120 ]
RT 60 FD 16 LT 60 FD 16 LT 120 FD 16
RT 60 FD 22 RT 120 FD 16 RT 60 FD 22
END
```

=====

**XB PROGRAMMING TIP**

I was working on a program and I ran into something interesting.

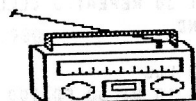
I had a variable (A) that could only be either 9 or 7. I needed to separate the beginning of a string. If A was 9, I needed the first 5 characters in the string, otherwise I needed the first 4 characters.

My first approach was:

```
IF A=9 THEN N$=SEG$(A$,1,5)
ELSE N$=SEG$(A$,1,4)
```

That seemed bulky so I tried this:

```
N$=SEG$(A$,1,5+(A=7))
```



This works because if A is 7, the expression (A=7) returns -1. Otherwise it returns 0. If this is not clear, try this loop:

```
FOR A=6 TO 8 ::
PRINT A;A=7;5+(A=7) ::
NEXT A
```

Note that the parenthesis are only needed for 5+(A=7). Without them, your TI would compare 5+A to 7.

Back to my tip. That was better, but I did a bit of fiddling and realized that 0.6 times 9 is 5.4 and 0.6 times 7 is 4.2. I tried this:

```
N$=SEG$(A$,1,INT(A*.6))
```

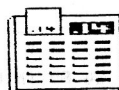
Then I wondered if the INT was needed. It turns out that SEG\$ (like many TI functions) automatically rounds off.

How can you tell if a function rounds? Try using it in a loop:

```
FOR I=1 TO 2 STEP .1
PRINT I;SEG$("ABC",1,I)
NEXT I
```

My final expressions was:

```
N$=SEG$(A$,1,A*.6)
```



**TREASURE TRAIL**

By Peter J Hutchison

Unfortunately I got no response from the first Trail in the last issue (there *MUST* be some Adventurers out there somewhere!) but never mind, if you are stuck or you have hints or solutions to adventures please send them to:

6 Moorlands View, Free School Lane, Savile Park, HALIFAX, HX1 2XQ

I have now completed The Count (a real pain in the neck to complete!) and here are a few tips for those stuck with it. I've had to play it from scratch in different orders to complete it!

Can't open the workroom door? tsop eht rof tiaW.  
 Can't climb down castle wall? deb eht esU.  
 The Bat is a pain? yrtnap eht enimaxE.  
 Can't go into the oven? thgilnus fo etisoppo eht stahW.  
 Can't find the crypt? wodniw moordeb eht hguorht oG.  
 Can't find the coffin? ngis eht daeR.  
 Loosing important objects? ecalp ytsud a ni meht kcol.  
 Falling asleep? ecalp ytsud a dniF.

Next issue I hope to pass on tips on Strange Odyssey when I finish it. I would love to hear from anybody who has written adventures in Basic (especially with graphics) and to hear of any techniques they have used to save memory, the parser, and so on.

[ Apologies for mis-spelling Peter's surname in the last issue, stated correctly above sjs].

-----

Request: Can anyone develop a bit-map mode Mandelbrot program for me in FAST machine code? The essential loop is short enough for at least a part to go into scratch pad ram. See what you can do. The rough layout is:

- input initial X and Y values (between 4 and -4)
- input length of SIDE (min 0 max 4)
- XCONST=X; FOR N=1 TO 192; XCONST=XCONST+SIDE/192
- YCONST=Y; FOR M=1 TO 256; YCONST=YCONST+SIDE/256
- XX=0; YY=0; C=0;
- repeat
  - XTEMP=XX\*XX; YTEMP=YY\*YY;
  - TEMP=XTEMP-YTEMP+XCONST
 (in another program use: TEMP=YTEMP-XTEMP+XCONST)  
 Y=2\*XX\*YY+YCONST; X=TEMP; C=C+1;
- until (C>100 OR XTEMP+YTEMP>4)
- plot point n,m- color determined by value of C,norm is black if c>100. Perhaps allow user to set colours for various values of C? Allow for color bleed in bitmap! At our resolution if colour values are fixed allow for change of colour every 10:-(INT(C/10))
- NEXT M; NEXT N;

For a third variation, allow user to set variable MELT and in line e above set XX=MELT; YY=MELT; where MELT is less than 1: this has an odd effect on an Archimedes, what can a TI do with it?

Fastish results (say a new screen every 20 minutes or less) to Stephen Shaw please. PLEASE program this for me!



**TEXAMENTS Micro Computer Specialists**

53 Center Street, Patchogue, New York 11772, (516)475-3480, BBS (516)475-6463

For more information contact:  
**Steven C. Lamberti**  
 (516)475-3480

**For Immediate Release**

TEXAMENTS NOW FORMING COLLECTION OF TI BASE MATERIAL

Patchogue, New York -- Since TI Base was introduced, users have been searching for information and tutorials beyond those originally included with the program. During this time, numerous articles regarding TI Base have appeared in MICROpendium and in various user group newsletters. Responding to the requests of many TI Base users, Texaments is now attempting to compile and organize these articles, and make them available free of charge from a convenient central location.

"TI Base is quickly being recognized as the new standard of database management within the TI community", explained Steve Lamberti, President of Texaments. "We're establishing this collection of TI Base material to further expand our support of TI Base. To help establish and support this collection of information, we are asking that anyone who has written an article, a command [program] file, or a utility of any kind, to send them to us for free public distribution."

TI Base users are encouraged to submit articles they have written about TI Base, command (program) files they have created, or program utilities they have produced. Anything submitted that is perceived to be valuable to others will be placed in a special TI Base Forum located on TI Source(tm). TI Source(tm), a free multi-user bulletin board system sponsored by Texaments, can be reached around the clock at (516)475-6463 by modem. Although the articles, command files, and utilities received will initially be available only through TI Source(tm), Texaments does intend to distribute these materials by mail in the future.

Whenever possible, submissions should be on floppy disk, or directly uploaded by modem to TI Source(tm). All submissions should be sent to the address below:

Texaments  
 Department TIBC  
 53 Center Street  
 Patchogue, NY 11772

Make sure that you include your name, address, and telephone number with any submissions or correspondence. Ownership, copyright, and any other claims made to the material submitted will be retained by the author unless otherwise agreed upon.

Article by Bruce and Dolores Harrison for TI \* MES:  
 Machine Code and Music:

NEW MUSIC FROM HARRISON SOFTWARE

Many people have done music programs on the TI-99/4A home computer, and many have done well at that, writing in Extended Basic. This story is about a new way of doing music programs in Assembly Language, and then making the music play as if it were Extended Basic. What we've done at Harrison Software has been a team effort between our resident musician and our resident assembly programmer. Neither could create this music alone.

The effort began a couple of years ago, when our musician (Dolores P. Werths) wanted to hear some music by Johann Christian Bach. That music, which is a set of six piano sonatas, is very rarely played, and was not available on records. She asked our resident assembly programmer (Bruce Harrison) for some help in getting the TI to play the music, so she could listen to it and study the compositions. After making some attempts in Extended Basic, which was really not satisfactory, the effort to do it all assembly was started.

The team effort was absolutely necessary because the assembly programmer cannot read music, and the musician cannot deal with all the intricacies of assembly, including the unfriendly Hexadecimal number system. The programmer then made up a file of nothing but equates called the "NOTES" file. This file allows the musician to create music in her own quasi-musical notation, while the assembler takes care of converting the musician's input data into that mysterious Hexadecimal stuff. The notes file allows the musician to enter such statements as:

```
DATA D32
BYTE V52
```

The musician means by this that a third-octave D is to be played by generator 2, and that volume level 5 is assigned to that generator. This concept, where all the available notes and volumes are entered as mnemonics is still in use in all our music efforts, but we had another hurdle to overcome. That hurdle is the matter of timing the durations of the notes when playing. That was important because in the Baroque and Galante music, there are numerous musical ornaments, and these require very fast and precise timing to sound right.

In the TI Editor/Assembler book, there is a whole chapter devoted to sound, and the method recommended by TI was to create "sound lists" in main memory and to put these bytes in VDP Ram, then let the VDP produce the sound through the generators. Unfortunately, the timing of sound durations by the VDP is done using its vertical interval as the clock. This means that the timing is all done in 60ths of a second, and that proved to be a killing limitation for the ornaments in the music. In other words, one cannot properly execute 64th note triplets when the smallest duration for either a note or a rest is 1/60th of a second.

Fortunately, there was another method of doing things in assembly, which was mentioned in passing by TI in that giant book. That is the method of passing sound bytes directly to the sound generator chip at address Hex 840. TI of course pointed out that when you do this, you must use your own timing loop to control the duration of the sound. This turned out to be the main ingredient of our success.

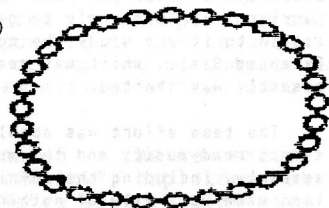


We soon found out that our timing loop, even with purposeful time-wasting operations built in, would have to execute over 200 times for a 64th note at a Presto pace. This gave us the flexibility to execute all the ornaments flawlessly.

To make all that accessible to the musician, we created the concept of the "TEMPO" file. This too, like NOTES, was all equates, and required only one numeric entry to define the duration of the 64th note, then let the assembler derive the durations for 32nd, 16th, 8th, and so on, from that initial entry. Even the durations for triplet notes are entered as simple math expressions, which the assembler computes for us. Again, as with the notes, the durations were translated into easy mnemonics, so that the musician enters DATA Q for a quarter note, DATA E for an eighth, and so on.

That file of duration equates, which we call the TEMPO or simply T file, looks like this:

TEMP	EQU	250	(RANGE 4 THRU 2000)
SX	EQU	TEMP	(64TH NOTE)
T	EQU	TEMP*2	(32ND NOTE)
S	EQU	TEMP*4	(16TH NOTE)
E	EQU	TEMP*8	(8TH NOTE)
Q	EQU	TEMP*16	(QUARTER NOTE)
H	EQU	TEMP*32	(HALF NOTE)
TS	EQU	E/3	(TRIPLET 16TH)
TT	EQU	S/3	(TRIPLET 32ND)



Thus all the durations are derived from a single programmer entry. Notice that the range of possible values for TEMP is very broad, but values below 200 make the music very fast indeed, while values above 400 make for truly "dirge" playing.

Our resident musician is very careful about the markings given on the sheet music, and sometimes takes many attempts at the tempo number before she's satisfied that the music is being played "Allegro Assai" and not "Presto", or vice versa.

This musical integrity is one of the things that we insist on, and it's yet another reason that our music is a team effort. Many of the music programs we've seen by others have all kinds of superb graphics and special effects of one kind or another, but they are lacking the musician's touch, in that the rhythms are inaccurate, or the various ornaments are improperly executed. For us, that detracts from the enjoyment of the music. In all fairness, many of these are actually programmed in basic or extended basic, so not all of the musical faults can be laid at the programmers' feet. We know from our own experience that it's sometimes impossible to get "CALL SOUND" to do what you really want.

Of course there's more to the musician's touch than just timing. There's also the matter of dynamics and musical balance to consider. The dynamics are of course not anywhere near the available range of real musical instruments. The total range available on the TI sound generators is only 30 decibels, and because of the way we use volume settings, we are limited to only 28 decibels. Even so, we can produce dramatic changes in volume. Our resident musician is very careful in the use of large swings in volume. Rarely do the scores call for maximum Forte or minimum Piano.

For that matter, some of the sheet music we use gives no indication of dynamics at all. In some of those cases, we can listen to recordings of the music played by first-rate musicians, and try to emulate their style. In the Nutcracker Suite, for example, many cues were taken from a recording by the Philadelphia Orchestra under Eugene Ormandy. Of course there are many recordings of that work which could be used for guidance. In the cases of such works as Johann Christian Bach's Sonatas, Opus V and XVII, there are no recordings readily available, so the musician must study other works by that composer very carefully, to get the sense of how he would wish to have his music played.

Having a musician "on staff" here has made all the difference in the music we create. On occasion, she even finds mistakes in the printed scores, and with her knowledge of music is able to correct them. Sometimes the publisher or editor has misinterpreted an ornament indication, or even put in a wrong note. Being able to detect and correct such errors makes our music sound better.

In much of the music we do here, the ornamentation is indicated simply by markings, and it is left to the performer to interpret the markings in a correct musical manner. Our musician has studied all this with great care, so that the actual playing of the ornaments is made smoothly and appropriate for its musical context. She often has to resort to working out the timings on graph paper, to make sure that the notes in the ornamentation work out properly timed with the longer duration notes in the bass line. One of the more challenging instances was in the Nutcracker Suite, where Tchaikovsky had a tied group of seven notes in the space of one quarter note. The timing for that had to be worked out on a grid where each square represented 1/28th note. Fortunately, once the timing had been worked out, the numbers could be expressed directly in the source file so that the assembler did the calculations of the actual durations. For example, if a note needs a duration of one seventh of a quarter note, we can simply enter Q/7 in a data statement, and let the assembler take the value assigned to Q and divide that by seven. Of course there is some "rounding error" in the calculation, since the assembler works only in integer numbers, but the effect is minimal because the numbers are large enough. The value for a quarter note is somewhere around 3800, so dividing by seven and leaving off the part beyond the decimal point is still well less than a one per cent error, and is not a noticeable difference when the music is playing.

That leads to another important point in the way we do our music. With our methodology we can make the computer play much faster than any human player could. We try very hard to keep within the "humanly possible" limits, including our interpretation of ornaments. If our musician feels that the computer is playing something that no human could match, she'll either slow down the tempo slightly or simplify the ornament so that it's within human capability. A good friend of ours who is a very advanced amateur flutist, and who also owns a TI, has been a "guinea pig" for much of our musical work. He provides some feedback on our musical aspects, and has often remarked on the "human" quality of our work, even though played by a machine.

Of course many of you still have some burning questions as to how this assembly language stuff gets to appear to the computer as if it were simple Extended Basic programs. This was not easy to accomplish the first time, but it's really not all that complicated.

In this example, we're showing what was done in one of our numbers from Anna Magdalena's Notebook, by Johann Sebastian Bach.

We use a bit of trickery which we call "submerging" assembly programs. First we assemble the program in the usual way with the E/A module. Next, we examine the first line of the object file, to discover the length of the program. That gives us a hexadecimal number, say for example >1DC4. We use that number to determine where the program needs to be loaded under an extended basic program. Extended Basic always loads XB programs in high memory, at a high enough address so that the area from >FFE7 through >FFFF is left open. What we then do is to subtract the length of the assembly program from >FFE6. For our example, >FFE6->1DC4 = >E222. We then go back to our source file and put in an Absolute Origin (AORG) directive at the beginning of the program, which we also insure is the entry point of that code. Now we re-assemble the program using that absolute origin. That's where the Loader will place the machine code in memory when we load this new object file.

Now the real tricky part begins. We take that address, subtract 16 from it to allow a small buffer zone in memory, giving us in this case >E212. We separate that number into its two bytes and convert each byte to a decimal number. Thus >E2 becomes 226 and >12 becomes 18.

Now we go into Extended Basic, do a CALL INIT, NEW, and then a CALL LOAD(-31952,226,18,226,18). This trick makes the Extended Basic interpreter behave as if the bottom of memory is already filled with program. Now we create a one-line program such as:

```
100 CALL LINK("AM7")
```

That one line program will serve as the link to our assembly code, but first we do the following in command (immediate) mode:

```
CALL LOAD("DSK1.ANMA67")
```

The Extended basic loader now loads in the assembly object file ANMA67 just after our 16 byte buffer into the higher locations in high memory. Now we simply do a save under another file name, and the machine code we just loaded in is saved to disk just as if it were part of that one-line program. All that would be fine except that the computer would not know how to execute that CALL LINK in line 100. That's where our second trick comes in.

In the LOAD program which puts the selection menu on the screen, we execute a CALL LOAD("DSK1.DEFAM"). That file DEFAM is a zero-length object file. That is, an assembly language program which contains no program as such. We use it merely as a vehicle to trick the loader into placing the addresses of all our 20 pieces into low memory at one whack. The source code for DEFAM looks like this:

```
DEF AM1,AM2,AM3,AM4
DEF AM5,AM6,AM7,AM8
(ETC), THEN
AM1 EQU >FCF4
AM2 EQU >FD4A
...
AM7 EQU >E222
```

And so on. Thus the absolute address for the assembly part of each piece is loaded along with its name in the lookup table, so that CALL LINK("AM7") causes the computer to branch directly to address >E222. That is why we insist that you run these music programs from their LOAD program, otherwise the computer will not know where to find them.

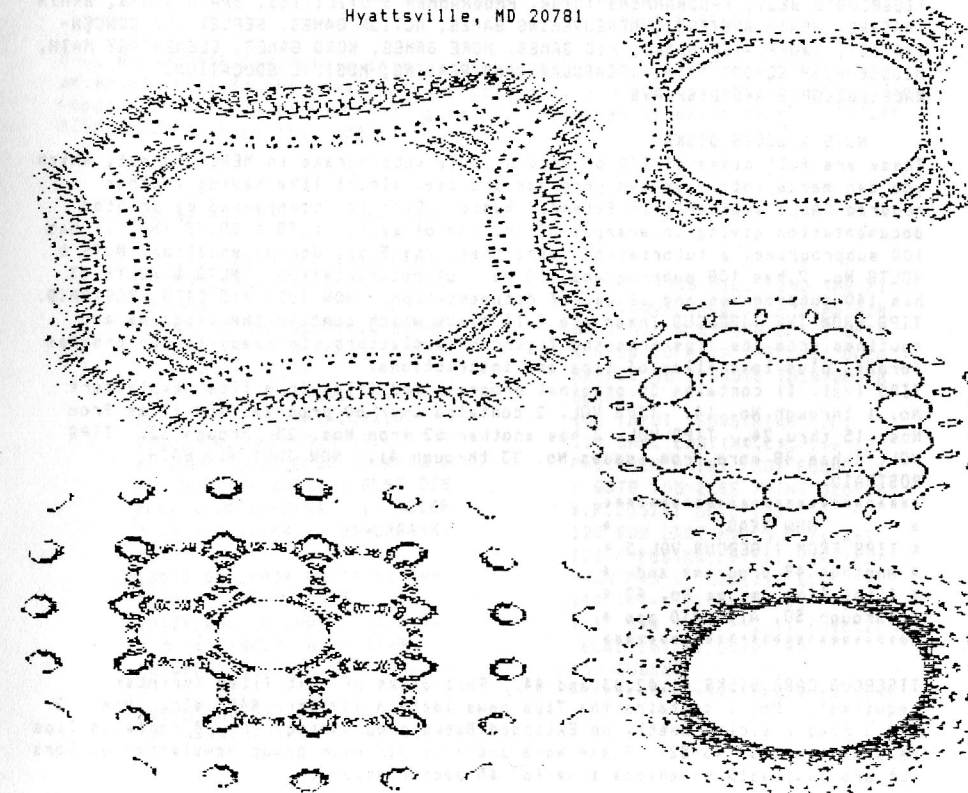
That leaves one more burning question to handle: How do we make the TI's sound generators behave like a piano, celesta, harp, and so on? Again the root of the process lies in the fact that we deal directly with the sound generators, and perform our own timing through a loop.

While that loop is executing, we can make it periodically reduce the volume on the generators that are active, and by tapering the time between these volume reductions, we can simulate the exponential decay of a struck piano string. We make different instruments by varying the overall time of the decay and the change of slope during that decay. The decay time is related to the tempo, so that in slower pieces the sustain is longer, while in faster pieces the faster decay prevents blurring of the sound. Of course the working out of the numbers for a new instrument effect takes some trial and error, but we can now produce a pretty wide variety of such effects.

The one instrument we're still struggling with is the violin. That instrument family is characterized by a triangular waveshape which is difficult to simulate with the TI's square wave generators. Stay tuned, though, we're getting closer all the time.

The bottom line to all this is that we have created a new kind of music on the TI, and we're thrilled that other people enjoy it enough to buy our software. If you'd like more information, please feel free to write us at:

5705 40th Place  
Hyattsville, MD 20781





TIPS FROM THE TIGERCUB  
#50 & #51 & #52

Copyright 1988  
TIGERCUB SOFTWARE  
156 Collingwood Ave. Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 120 original programs in Basic and Extended Basic, available on cassette or disk, NOW REDUCED TO JUST \$1.00 EACH!, plus \$1.50 per order for cassette or disk and PPM. Minimum order of \$10.00. Cassette programs will not be available after my present stock of blanks is exhausted. The Handy Dandy series, and Color Programming Tutor, are no longer available on cassette. Descriptive catalogs, while they last, \$1.00 which is deductible from your first order.

Tigercub Full Disk Collections, reduced to \$5 postpaid. Each of these contains either 5 or 6 of my regular catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus! TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION GAMES, REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPES AND DISPLAYS

NUTS & BOLTS DISKS

These are full disks of 100 or more utility subprograms in MERGE format, which you can merge into your own programs and use, almost like having another hundred CALLs available in Extended Basic. Each is accompanied by printed documentation giving an example of the use of each. NUTS & BOLTS (No. 1) has 100 subprograms, a tutorial on using them, and 5 pp. documentation. NUTS & BOLTS No. 2 has 108 subprograms, 10 pp. of documentation. NUTS & BOLTS #3 has 140 subprograms and 11 pp. of documentation. NOW JUST \$15 EACH, POSTPAID. TIPS FROM THE TIGERCUB These are full disks which contain the programs and routines from the Tips from the Tigercub newsletters, in ready-to-run program format, plus text files of tips and instructions.

TIPS (Vol. 1) contains 50 original programs and files from Tips newsletters No. 1 through No. 14. TIPS VOL. 2 contains over 60 programs and files from Nos. 15 thru 24. TIPS VOL. 3 has another 62 from Nos. 25 through 32. TIPS VOL. 4 has 48 more from issues No. 33 through 41. NOW JUST \$10 EACH, POSTPAID.

```
*****
*           NOW READY           *
* * TIPS FROM TIGERCUB VOL.5 * *
* * Another 49 programs and * *
* * files from issues No. 42 * *
* * through 50. Also $10 ppd * *
*****
```

TIGERCUB CARE DISKS #1,#2,#3 and #4. Full disks of text files (printer required). No. 1 contains the Tips news letters #42 thru #45, etc. Nos. 2 and 3 have articles mostly on Extended Basic programming. No. 4 contains Tips newsletters Nos. 46-52. These were prepared for user group newsletter editors but are available to anyone else for \$5 each postpaid.

Some folks seem to think that the subprograms on my Nuts & Bolts disks are just flashy screen displays. Not so! This one will be on the next diskfull, if I ever get it full, which is most unlikely.

ACCEPT AT with a negative size is useful to accept a default string from the screen, but the length of the string is limited to 28 characters; and if you want something other than the default, you must be sure to delete any extra characters. CALL DEFAULT(R,C,M\$,R\$), where R and C are the row and column to accept at, M\$ is the default string which can be up to 254 characters long, and R\$ is the string accepted, will display the default string, accept it if Enter is pressed, or accept any other string without having to blank out the extra characters. Just don't type too fast!

```
100 M$="TESTING" :: CALL CLE
AR
110 CALL DEFAULT(12,1,M$,R$)
:: DISPLAY AT(24,1):R$ :: GO
TO 110
10000 SUB DEFAULT(R,C,M$,R$)
:: R$="" :: X=ASC(M$)
10001 DISPLAY AT(R,C):M$
```

```
10002 CALL HCHAR(R,C+2,ASC(S
EG$(M$,1,1))): CALL HCHAR(
,C+2,30)
10003 CALL KEY(0,K,S):: IF S
=0 THEN 10002 ELSE IF K=13 T
HEN R$=M$ :: SUBEXIT ELSE DI
SPLAY AT(R,C):CHR$(K):: ACCE
PT AT(R,C+1):R$ :: R$=CHR$(K
)&R$
10004 SUBEND
```

CALL DEFAULTN(R,C,N,RN), with N as the default value and RN as the value accepted, will do the same for numeric input, and will reject any non-numeric input. Errors due to fast typing can be prevented by omitting the DISPLAY AT(R,C):CHR\$(K) in line 1002.

```
100 N=176453.897 :: CALL CLE
AR
110 CALL DEFAULTN(12,1,N,RN)
:: DISPLAY AT(24,1):RN :: GO
TO 9999
10000 SUB DEFAULTN(R,C,N,RN)
:: DISPLAY AT(R,C):N :: N$=S
EG$(STR$(N),1,1)
10001 CALL HCHAR(R,C+2,ASC(N
$)): CALL HCHAR(R,C+2,30)
10002 CALL KEY(0,K,S):: IF S
=0 THEN 10001 ELSE IF K=13 T
HEN RN=N :: SUBEXIT ELSE DIS
PLAY AT(R,C):CHR$(K):: ACCEP
T AT(R,C+1):R$ :: R$=CHR$(K)
&R$
10003 ON ERROR 10004 :: RN=V
AL(R$):: GO TO 10005
10004 CALL SOUND(200,110,5,-
4,5):: DISPLAY AT(R,C):N ::
ON ERROR STOP :: RETURN 1000
2
10005 SUBEND
```

```
100 M$="POS WILL FIND THE FI
RST OCCURRENCE OF A SUBSTRIN
G WITHIN A STRING BUT I OFTE
N NEED TO FIND THE LAST OCCU
RRENCE SO I WROTE THIS SUBPR
OGRAM"
105 INPUT "SUBSTRING?":L$
110 CALL LAST(M$,L$,P):: IF
P=0 THEN PRINT "NOT FOUND" :
: GOTO 105 ELSE PRINT SEG$(M
$,P,255):: GOTO 105
120 SUB LAST(M$,L$,P):: X=1
130 Y=POS(M$,L$,X):: IF Y=0
THEN P=0 :: SUBEXIT ELSE Z=Y
140 X=Y+1 :: Y=POS(M$,L$,X):
: IF Y=0 THEN P=Z :: SUBEXIT
ELSE Z=Y :: GOTO 140
150 SUBEND
```

When one program is run from another by RUN DSK... the screen is not cleared, sprites are not deleted, and screen color, character definitions and sprite magnification are not returned to the default values. This can cause some strange results, which can be prevented by CALLING CLEARALL just before the RUN.

```

1000 SUB CLEARALL :: CALL CL
EAR :: CALL DELSPRITE(ALL)::
CALL SCREEN(B):: CALL CHARS
ET :: CALL MAGNIFY(1)
1001 FOR CH=65 TO 90 :: CALL
CHARPAT(CH,CH):: CALL CHAR
(CH+32,"00"&SEG$(CH$,1,12)&
EG$(CH$,15,2)):: NEXT CH
1002 CALL CHAR(96,"000201008
",123,"0018202040202018",124
,"00101010001010100030080804
08083000000205408")
1003 FOR CH=127 TO 143 :: CA
LL CHAR(CH,"0"):: NEXT CH ::
SUBEND
    
```

Here are three screen display subprograms of the type you will find on my Nuts and Bolts disks. Note that subprograms can read DATA from the main program. The double commas in the DATA statement cause input of null strings of data for spacing between the lines. The M\$( ) in the subprogram parameter lists is necessary, even though the array is not passed from the main program, in order to DIMension the array in the subprogram - unless you prefer to place the DIM in the subprogram itself. T is the number of DATA items to be read.

```

100 CALL CLEAR
110 DATA THIS IS A DEMO,,OF
THREE SCREEN PRINTING,,SUBPR
OGRAMS PUBLISHED IN,,TIPS FR
OM THE TIGERCUB,,No. 51,,BY
TIGERCUB SOFTWARE
120 DIM M$(11):: CALL DOWNPR
INT(M$( ),11):: FOR D=1 TO 10
00 :: NEXT D :: CALL CLEAR :
: RESTORE 110 :: CALL DIAGPR
INT(M$( ),11)
130 FOR D=1 TO 1000 :: NEXT
D :: CALL CLEAR :: RESTORE 1
10 :: CALL INWARD(M$( ),11)
1000 SUB DOWNPRINT(M$( ),T)
1001 FOR J=1 TO T :: READ M$
(J):: L=INT(LEN(M$(J))+.5)::
M$(J)=RPT$(" ",14-INT(L/2))
&M$(J):: M$(J)=M$(J)&RPT$("
",28-LEN(M$(J))):: NEXT J
1002 FOR J=1 TO 28 :: FOR L=
1 TO T
1003 DISPLAY AT(L,1):SEG$(M$
(L).1,J):: NEXT L
1004 NEXT J :: SUBEND
2000 SUB INWARD(M$( ),T):: FO
R J=1 TO T :: READ M$(J):: N
EXT J :: R=1 :: FOR A=1 TO T
2001 L=INT(LEN(M$(A))):: F=1
3-L/2 :: G=L+F
2002 FOR J=1 TO INT(L/2+.5)::
DISPLAY AT(R,F+1):SEG$(M$(
A),J,1):: DISPLAY AT(R,G):S
EG$(M$(A),L-J+1,1):: F=F+1
:: G=G-1 :: NEXT J :: R=R+1
:: NEXT A :: SUBEND
3000 SUB DIAGPRINT(M$( ),T)::
FOR J=1 TO T :: READ M$(J)::
L=INT(LEN(M$(J))+.5):: M$(
J)=RPT$(" ",14-(L/2))&M$(J)::
M$(J)=M$(J)&RPT$(" ",28-LE
N(M$(J))):: NEXT J
3001 FOR J=1 TO 28+L :: FOR
L=1 TO T
3002 IF J<L THEN 3007
3003 DISPLAY AT(L,1):SEG$(M$
(L).1,J-L):: NEXT L
3004 NEXT J :: SUBEND
    
```

Just in case you didn't know - to jump directly to the first or last line in a TI-Writer file, use FCTN 9 and S(earch) and 0 for the first line or E for the last.

Here's a new way to make music. The algorithm in 110 sets up a 3-octave chromatic scale - note the N(1)=F, I have erroneously omitted it when I previously published that algorithm.

To change the key of the music you have programmed, just change the value of F. Lines 190-220 contain the part of the music that is repeated within the melody. A is the subscript of the melody note, B is the subscript number of the chord. These must be above 13, as the frequency is divided by 2 in the subroutine.

Each beat of the music has a GOSUB, to 230 to play a bass accompaniment with the first note of each bar, to 260 for the other notes of the bar. The chord note is divided by different values to play the three notes of the chord in succession, and multiplied by 3.75 in the 3rd voice to produce a bass note two octaves lower in the -4 noise. The melody note is multiplied by 1.01 in the second voice to give a richer tone.

```

100 DISPLAY AT(12,3)ERASE AL
L:"THE MAORI FAREWELL SONG"
! programmed by
Jim Peterson
110 F=110 :: DIM N(36):: FOR
J=1 TO 36 :: N(J)=INT(F*1.0
59463094^(J-1)):: NEXT J ::
N(1)=F :: T=-999
120 GOSUB 190 :: A=30 :: B=2
3 :: GOSUB 230 :: GOSUB 260
:: GOSUB 260 :: A=32 :: B=28
:: GOSUB 230 :: GOSUB 260 :
: GOSUB 260 :: A=28
130 GOSUB 230 :: GOSUB 260 :
: GOSUB 260 :: A=30 :: B=23
:: GOSUB 230 :: GOSUB 260 ::
A=28 :: GOSUB 260 :: A=27 :
: GOSUB 230 :: GOSUB 260
140 A=28 :: GOSUB 260 :: A=3
0 :: GOSUB 230 :: GOSUB 260
:: GOSUB 260 :: GOSUB 230 :: G
OSUB 190
150 A=30 :: B=23 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
:: A=32 :: B=16 :: GOSUB 230
:: GOSUB 260 :: A=28 :: GOS
UB 260
160 A=33 :: B=23 :: GOSUB 23
0 :: GOSUB 260 :: A=32 :: GO
SUB 260 :: A=25 :: B=13 :: G
OSUB 230 :: GOSUB 260 :: GOS
UB 260
170 A=27 :: B=23 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
:: A=28 :: B=16 :: GOSUB 230
:: GOSUB 260 :: GOSUB 260
180 B=28 :: GOSUB 230 :: GOS
UB 260 :: GOSUB 260 :: B=16
:: GOSUB 230 :: GOSUB 260 ::
GOSUB 260 :: GOTO 120
190 A=32 :: B=27 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
:: A=28 :: B=16 :: GOSUB 230
:: GOSUB 260 :: A=30 :: GOS
UB 260
200 A=32 :: B=28 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
:: B=16 :: GOSUB 230 :: GOSU
B 260 :: B=28 :: GOSUB 260 :
: GOSUB 230 :: GOSUB 260
210 A=30 :: GOSUB 260 :: A=3
3 :: B=23 :: GOSUB 230 :: GO
SUB 260 :: A=27 :: GOSUB 260
:: A=28 :: B=16 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
220 B=28 :: GOSUB 230 :: GOS
UB 260 :: GOSUB 260 :: B=16
:: GOSUB 230 :: GOSUB 260 ::
GOSUB 260 :: RETURN
230 CALL SOUND(T,N(A),5,N(B)
/1.585,9,N(B)*3.75,30,-4,9)::
GOSUB 290
240 CALL SOUND(T,N(A),5,N(B)
/1.334,9,N(B)*3.75,30,-4,9)::
GOSUB 290
250 CALL SOUND(T,N(A),5,N(B)
/2,9,N(B)*3.75,30,-4,9):: GO
SUB 290 :: RETURN
260 CALL SOUND(T,N(A),5,N(A)
*1.01,5,N(B)/1.585,9):: GOSU
B 290
270 CALL SOUND(T,N(A),5,N(A)
*1.01,5,N(B)/1.334,9):: GOSU
B 290
280 CALL SOUND(T,N(A),5,N(A)
*1.01,5,N(B)/2,9)
290 FOR D=1 TO 20 :: NEXT D
:: RETURN
    
```

MEMORY FULL.....

Jim Peterson



```

100 REM *****
110 REM * QUINTUS *
120 REM *****
130 REM BY SAM PINCUS
140 REM 99'ER VERSION 2.5.1
150 REM
160 RANDOMIZE
170 DIM GR(6,6)
180 GOSUB 1820
190 X=1
200 Y=1
210 J=2*Y+3
220 I=2*X+3
230 CALL GCHAR(I,J,CH)
240 Z=0
250 CALL HCHAR(I,J,100,2)
260 CALL HCHAR(I+1,J,100,2)
270 CALL KEY(1,R,S)
280 IF S THEN 360
290 IF Z THEN 240
300 Z=1
310 CALL HCHAR(I,J,CH)
320 CALL HCHAR(I+1,J,CH)
330 CALL HCHAR(I,J+1,CH)
340 CALL HCHAR(I+1,J+1,CH)
350 GOTO 270
360 IF CH<>96 THEN 420
370 CALL HCHAR(I,J,96)
380 CALL HCHAR(I+1,J,98)
390 CALL HCHAR(I,J+1,97)
400 CALL HCHAR(I+1,J+1,99)
410 GOTO 440
420 CALL HCHAR(I,J,CH,2)
430 CALL HCHAR(I+1,J,CH,2)
440 IF R=2 THEN 590
450 IF R=3 THEN 630
460 IF R=5 THEN 510
470 IF R+1=1 THEN 550
480 IF R=1 THEN 670
490 GOTO 240
500 REM UP
510 IF X=1 THEN 290
520 X=X-1
530 GOTO 210
540 REM DOWN
550 IF X=5 THEN 290
560 X=X+1
570 GOTO 210
580 REM LEFT
590 IF Y=1 THEN 290
600 Y=Y-1
610 GOTO 210
620 REM RIGHT
630 IF Y=5 THEN 290
640 Y=Y+1
650 GOTO 210
660 REM HIT
670 IF GR(X,Y)=1 THEN 1370
680 IF GR(X,Y)=3 THEN 1370
690 TU=1

```

```

700 IF GR(X-1,Y)>1 THEN 720
710 GR(X-1,Y)=GR(X-1,Y)+2
720 IF GR(X+1,Y)>1 THEN 740
730 GR(X+1,Y)=GR(X+1,Y)+2
740 IF GR(X,Y-1)>1 THEN 760
750 GR(X,Y-1)=GR(X,Y-1)+2
760 IF GR(X,Y+1)>1 THEN 780
770 GR(X,Y+1)=GR(X,Y+1)+2
780 GR(X,Y)=3
790 HSC1=HSC1+1
800 CALL HCHAR(I,J,104,2)
810 CALL HCHAR(I+1,J,104,2)
820 IF SC=-100 THEN 1290
830 A=1
840 B=1
850 RO=21
860 CO=12
870 MS#="MY TURN"
880 GOSUB 2490
890 IF GR(3,3)>1 THEN 930
900 A=3
910 B=3
920 GOTO 1150
930 SC=-100
940 FOR C=1 TO 5
950 FOR D=1 TO 5
960 IF GR(C,D)>1 THEN 1120
970 TSC=0
980 XY=GR(C-1,D)
990 TSC=TSC-1.5*(XY=0)-2*(XY
=2)+2*(XY=3)
1000 XY=GR(C+1,D)
1010 TSC=TSC-1.5*(XY=0)-2*(X
Y=2)+2*(XY=3)
1020 XY=GR(C,D-1)
1030 TSC=TSC-1.5*(XY=0)-2*(X
Y=2)+2*(XY=3)
1040 XY=GR(C,D+1)
1050 TSC=TSC-1.5*(XY=0)-2*(X
Y=2)+2*(XY=3)
1060 IF TSC<SC THEN 1120
1070 IF TSC>SC THEN 1090
1080 IF RND>=.5 THEN 1120
1090 SC=TSC
1100 A=C
1110 B=D
1120 NEXT D
1130 NEXT C
1140 IF SC=-100 THEN 1270
1150 GR(A,B)=3
1160 IF (GR(A-1,B)=0)+(GR(A-
1,B)=2) THEN 1170 ELSE 1180
1170 GR(A-1,B)=GR(A-1,B)+1
1180 IF (GR(A+1,B)=0)+(GR(A+
1,B)=2) THEN 1190 ELSE 1200
1190 GR(A+1,B)=GR(A+1,B)+1

```

```

1200 IF (GR(A,B-1)=0)+(GR(A,
B-1)=2) THEN 1210 ELSE 1220
1210 GR(A,B-1)=GR(A,B-1)+1
1220 IF (GR(A,B+1)=0)+(GR(A,
B+1)=2) THEN 1230 ELSE 1240
1230 GR(A,B+1)=GR(A,B+1)+1
1240 TISC1=TISC1+1
1250 CALL HCHAR(2*A+3,2*B+3,
112,2)
1260 CALL HCHAR(2*A+4,2*B+3,
112,2)
1270 MS#=""
1280 GOSUB 2490
1290 FOR A=1 TO 5
1300 FOR B=1 TO 5
1310 IF GR(A,B)>2 THEN 1340
1320 IF GR(A,B)=1 THEN 1340
1330 GOTO 190
1340 NEXT B
1350 NEXT A
1360 IF SC=-100 THEN 1400 EL
SE 820
1370 CALL SOUND(300,440,0,66
0,0,880,0)
1380 GOTO 240
1390 REM END OF GAME
1400 TSC=HSC1-TISC1
1410 IF TSC>0 THEN 1480
1420 IF TSC<0 THEN 1450
1430 MS#=" A TIE GAME"
1440 GOTO 1500
1450 TISC=TISC+ABS(TSC)
1460 MS#="I WON!!"
1470 GOTO 1500
1480 HSC=HSC+TSC
1490 MS#="YOU WON"
1500 RO=21
1510 CL=7
1520 GOSUB 2490
1530 GOSUB 2440
1540 IF TISC>=7 THEN 1580
1550 IF HSC>=7 THEN 1580
1560 GOSUB 2090
1570 GOTO 190
1580 RO=21
1590 CO=7
1600 MS#="GAME'S OVER "
1610 GOSUB 2490
1620 RO=22
1630 IF HSC>=7 THEN 1670
1640 CO=9
1650 MS#="I WON!"
1660 GOTO 1690
1670 MS#="YOU WON!"
1680 CO=8
1690 GOSUB 2490

```

```

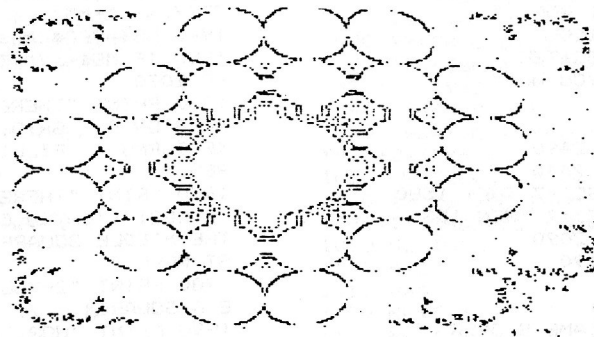
1700 GOSUB 2440
1710 GOSUB 2440
1720 CALL CLEAR
1730 INPUT "WANT TO PLAY AGA
IN? (Y/N) ":MS#
1740 MS#=SEG$(MS#,1,1)
1750 IF MS#="Y" THEN 1770
1760 STOP
1770 GOSUB 2070
1780 GOTO 190
1790 REM
1800 REM
1810 REM START OF PGM
1820 CALL CLEAR
1830 PRINT "          QUINTUS!
          BY SAM PINC
US": : : : : :
1840 CALL CHAR(96,"FF808080
808080")
1850 CALL CHAR(97,"FF010101
010101")
1860 CALL CHAR(98,"80808080
808080")
1870 CALL CHAR(99,"01010101
010101")
1880 CALL CHAR(100,"FFFFFFF
FFFFFFF")
1890 CALL CHAR(104,"FFFFFFF
FFFFFFF")
1900 CALL CHAR(112,"FFFFFFF
FFFFFFF")
1910 CALL COLOR(10,5,2)
1920 CALL COLOR(11,7,2)
1930 INPUT "NEED INSTRUCTION
S?(Y/N) ":MS#
1940 MS#=SEG$(MS#,1,1)
1950 IF MS#="Y" THEN 1960 EL
SE 2070
1960 PRINT "THERE ARE 25 SQU
ARES ON A GRID. WE BOTH TA
KE TURNS FILLING THE SQUA
RES."
1970 PRINT "THERE ARE ONLY 2
RULES:": "1-YOU CANNOT TAKE
THE MIDDLE SQUARE ON THE FIR
ST MOVE."
1980 PRINT "2-YOU CANNOT TAK
E A SQUARE"
1990 PRINT "HORIZONTALLY OR
VERTICALLY NEXT TO A SQUARE
THAT I OWN. MY SQUARES ARE
RED, YOUR"
2000 PRINT "SQUARES ARE BLUE
. THE CURSOR"
2010 PRINT "IS BLACK. USE TH
E 4 ARROW KEYS TO MOVE THE
CURSOR. PRESS 'A' TO CLA
IM A SQUARE."

```



```

2020 PRINT "A ROUND IS OVER
WHEN ALL OF THE POSSIBLE SQU
ARES ARE"
2030 PRINT "TAKEN. WHEN A RO
UND ENDS, THE WINNER RECEI
VES POINTS EQUAL TO THE AMO
UNT OF"
2040 PRINT "SQUARES HE WON B
Y. THE GAME IS OVER WHEN SOM
EDNE GETS 7 POINTS."
2050 PRINT "PRESS ANY KEY TO
START"
2060 GOSUB 2440
2070 TISC=0
2080 HSC=0
2090 CALL SCREEN(2)
2100 CALL CLEAR
2110 CALL COLOR(9,2,2)
2120 PRINT TAB(20);"HUMAN":
:TAB(21);HSC: : : :TAB(19);"
TI-99/4A": :TAB(21);TISC
2130 PRINT : : : : : : : :
: : : : :
2140 FOR X=5 TO 14 STEP 2
2150 CALL HCHAR(X,5,96,10)
2160 CALL HCHAR(X+1,5,98,10)
2170 CALL HCHAR(X,6,97)
2180 CALL HCHAR(X,8,97)
2190 CALL HCHAR(X,10,97)
2200 CALL HCHAR(X,12,97)
2210 CALL HCHAR(X,14,97)
    
```



```

2220 CALL HCHAR(X+1,14,99)
2230 CALL HCHAR(X+1,12,99)
2240 CALL HCHAR(X+1,10,99)
2250 CALL HCHAR(X+1,8,99)
2260 CALL HCHAR(X+1,6,99)
2270 NEXT X
2280 CALL SCREEN(12)
2290 CALL COLOR(9,2,8)
2300 HSC=0
2310 TISC=0
2320 TU=0
2330 SC=0
2340 FOR X=1 TO 5
2350 GR(X,0)=88
2360 GR(X,6)=88
2370 GR(0,X)=88
2380 GR(6,X)=88
2390 FOR Y=1 TO 5
2400 GR(X,Y)=0
2410 NEXT Y
2420 NEXT X
2430 RETURN
2440 FOR X=1 TO 1000
2450 CALL KEY(0,KEY,STAT)
2460 IF STAT<>0 THEN 2480
2470 NEXT X
2480 RETURN
2490 FOR X=1 TO LEN(MS$)
2500 CALL HCHAR(RO,CO+X-1,AS
C(SEG$(MS$,X,1)))
2510 NEXT X
2520 RETURN
    
```

SECTOR SHARING.

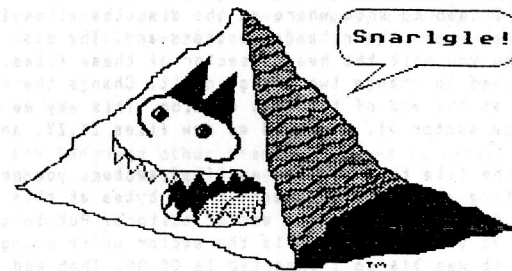
By Mark Shafer  
Bluegrass 99 Computer Society Inc.  
Retyped by S Shaw.

[ This is a complex but unusual article, addressing an odd problem in an odd way. Lateral thinking isn't in it.... sjs]

They say necessity is the mother of invention. And in this case, I'm the father. I think I've discovered something you'll find intriguing. Take a look at the following disk catalogue:

```

DSK1 - DISKNAME= FNWEB/4*1
AVAILABLE=117 USED= 241
FILENAME SIZE TYPE P
-----
AS          33 PROGRAM
AT          22 PROGRAM
CF          31 PROGRAM
CG          25 PROGRAM
D1          33 PROGRAM
D2          33 PROGRAM
D3          29 PROGRAM
DU          33 PROGRAM
DV          33 PROGRAM
DW          29 PROGRAM
LOAD       31 PROGRAM
    
```



If you don't notice anything strange, add up the sizes of the files and compare that to the number of sectors used. How did I do that? Why did I do that? That's what I'm here to tell you. What I've done is to make it so that some files take up the same space as other files, which is the concept I call secotr-sharing. First, let's get into why I did it. I have Disk Utilities by John Birdwell. One of its features is the ability to change to default system setup. The trouble was sometimes I will want the defaults to be one way and sometimes I will want them another. Now, I could change the setup in the program when I need to, but this is some trouble. The ideal solution would be to have two (or more) copies of the program on the disk, and boot the one with the defaults I want at the time. But I only have one SSSD disk drive, so I clearly don't have the room to do this if I want to add to the above disk the Funlweb files CHARA1, EA,ED, EE,QD, SL,SYSCON and UL! Just like limited memory can lead to tight coding, limited disk space can lead to creative disk utilisation. All I wanted to do was to change the first sector. So I got the idea to create a file that would have a different first sector, but share the rest of the sectors with the original file! The steps to do this, I believe, can be done in any order. Basically it goes like this: creating the new header sectors, creating the modified sector, updating the disk catalog, marking the used sectors, and renaming the new files.

CONTINUED--->



The beauty is that Disk Utilities itself can handle all of the above in one session, but I suppose any sector editor and disk manager will do. So let's create the new header sectors first. Each file on a disk has to have a sector that identifies the type of file it is and where it is on the disk. The first step here is to find out what sectors are free. One way to do this is to look at sector 0 starting at byte >38, look for non-F's, and figure out what sectors correspond to the blank bits.

Or you could use Disk Utilities to print the disk report and figure out what sectors are contained in no file. In my case, sectors >13 to >16 were available. You could put them anywhere, but the normal thing is to put header sectors in the >02 to >21 range.

Next I need to know where on the disk the files I'm going to "copy" are, as well as where their header sectors are. The disk report has this information. So now you edit the header sector of these files. So to the first one first. You need to change two things on it. Change the name to something that would fall at the end of the disk catalog. This way we don't have to insert when we change sector >1. I called my new files ZX, ZY, and ZZ.

For the file that has the modified sector, you need to change the segments starting at byte >1C. Insert three bytes at this point. This may be a little more difficult with some sector editors. Put in the following three bytes at >1C: yz 0x 00 where xyz is the sector we're going to create in step 2. In my case it was >16 so I inserted 16 00 00. Then add one to the next byte, so if it's >57, make it >58.

This process makes it so this file is in the same place as the original file except its first sector is different. If you're changing a sector in the middle, this is a bit more difficult. When you save it back, put it at the first available sector you found. For the remaining header sectors, I just simply changed their names and saved them to the next available sectors since they are to share exactly the same sectors.

Once you've got that done, the rest of it is a cinch. To create the modified sector, simply edit the sector you wish to change, make the appropriate changes, and save it to the free sector you indicated at step one (eg >16 in my case). Normally the sectors contained in a file are higher than >21 but I didn't have any free in that area.

Next it's time to change the disk catalog at sector 1. Simply put the header sectors you created at step one at the first available 0000 in sector one. I appended 0013 0014 0015 to add my three new files.

The next step is to tell the disk what sectors we've used. With Disk Utilities, you just use the Mark Sector feature. With others, you may have to figure out what bits they correspond to in sector 0 and make the changes yourself.

The last step is to rename the new files what you really want to call them. I called mine D1, D2, and D3. The last bytes have to be consecutive so that they load as one continuous program. After this, I had to configure Funweb to be able to load my new program. So now, when I run Disk Utilities, I have the choice of the options in DU or the options in D1. They both load just fine.

But there are some consequences. There's the problem of copying. If you try to copy a sector-sharing disk by file, the duplicate will unshare them! Also, you may get an out-of-space error. So to copy such a disk you should use a sector [or track-sjs] copier. Then there's the problem of what happens if you want to copy the sector sharing files but not the whole disk- if you can't do a direct copy sector x to sector x, I would recommend that you find a way to sector copy the whole disk, and delete the files you didn't want. Or you could copy only one of them and start the operation over again on the new disk.

CONTINUED.....->

So to make a copy of this 95 sector program, it took only four additional sectors! Three for each new header sector, and one for the modified sector. Shorter files would need even fewer additional sectors. I could go on and make another version of this program, but I think I'm happy with just two. I wonder if I have any other files I can do this to...

[[ Disk Utilities is available from the User Group library. If you can follow this by no means simple article you will quickly learn how to use Disk Utilities - which does have simpler applications - and you will certainly learn very effectively how your disk system works- we have already covered sector layouts in previous issues, this article really hammers it all home! sjs]]

PROGRAMMING HINT- NOT BASIC...

This note is appropriate to any language other than Basic- and is prompted by a member having problems with the command language of TI BASE.

His code was along the lines of:

```
WHILE RESPONSE="Y"
  INPUT PROCESS
  WRITE X,Y "ANOTHER?"
  READSTRING RESPONSE
  IF RESPONSE="N"
    CLOSE DATA
    RETURN
  ENDIF
ENDWHILE
```

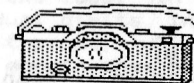


Repeated processing of this type of code was resulting in an error message 'too many if while or case levels'.

How too many levels- only one surely? One WHILE/ENDWHILE and one IF/ENDIF... no? NO! Actually too many whiles under certain circumstances!

Basic is unique in computer languages in that it allows you to get away with code like this:

```
100 FOR T=1 TO 1000
110 CALL KEY(I,A,B)
120 IF B=1 THEN 140
130 NEXT T
140 PRINT "DONE";T
```



in any other language- including the TI Base command language, this would spell disaster. The general rule of programming is that you do not jump out of a conditional loop until the condition is met- GOSUB can provide us with a Basic example:

```
100 GOSUB 1000
110 END
1000 FOR T=1 TO 1000
1010 CALL KEY(I,A,B)
1020 IF B=0 THEN 100
1030 NEXT T
1040 RETURN
```



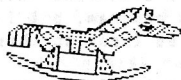
CONTINUED----->



if you run this, you will after a little while of not pressing any keys, receive the Basic equivalent message to the one you are getting with TI Base- each GOSUB is remembering where you jumped from, and unless you exit with a RETURN, you just keep remembering more and more return addresses until...

(I really have seen quite a lot of programs coded like that last bit of code!)

```
The BASIC program, in a format other languages can use, is:
100 FOR T=1 TO 600
110 CALL KEY(1,A,B)
120 IF B=1 THEN T=600
130 NEXT T
140 PRINT "DONE";T
```

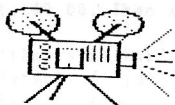


Basic actually clears the T flag when you jump out- other languages wont, they will keep waiting for the condition to be met, and if the code is processed frequently, memory will quickly go!

That Basic allows the first form, and teaches a method inapplicable to any other form of programming, has resulted in Basic receiving a lot of criticism from professional programmers. TI went a long way to meeting criticisms by offering in ExBas sub-programs with local variables- long before Acorn thought of it!

The first bit of code should be configured as below- we do not jump out of the conditional WHILE loop until the condition is met..

```
WHILE RESP="Y"
  INPUT PROCESS
  WRITE X,Y "ANOTHER?"
  READSTRING RESP
ENDWHILE
CLOSE DATA
RETURN
```



If you compare this to the first form, you will see it does the same thing, but actually uses the test in the WHILE. And as we only leave the WHILE loop when the condition is met, you dont eat up memory by nesting WHILE loops when you don't need to! (The code is also shorter).

This is one case where early experience with TI Logo would have helped, as that would rapidly have taught the perils of iteration ( thats the technical name for going around in circles!). TI LOGO actually counts the "level"s as you go down...

contd. from p.5

```
100 DIM U$(19)
110 FOR J=0 TO 19 :: READ U$(J):: NEXT J
120 DATA "",ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,ELEVEN,TWELVE,THIRTEEN,FOURTEEN,FIFTEEN
130 DATA SIX+TEEN,SEVEN+TEEN,EIGHT+TEEN,NINE+TEEN
140 FOR J=2 TO 9 :: READ TN$(J):: NEXT J
150 DATA TWENTY,THIRTY,FORTY,FIFTY,SIXTY,SEVENTY,EIGHTY,NINETY
160 CALL CLEAR
170 SAY$=""
180 INPUT "NUMBER ":N
```

```
190 IF N>999 THEN 180
200 IF N=0 THEN CALL SAY("0"):: GOTO 180
210 H=INT(N/100):: IF H>0 THEN SAY$=U$(H)+"HUNDRED"
220 N=N-H*100 :: IF N<>0 AND H<>0 THEN SAY$=SAY$+"AND"
230 TN=INT(N/10):: UN=N-TN*10
240 IF TN=0 THEN SAY$=SAY$+"&U$(UN):: GOTO 270
250 IF TN=1 THEN SAY$=SAY$+"&U$(UN+10):: GOTO 270
260 SAY$=SAY$+"&TN$(TN)+"&U$(UN)
270 CALL SAY(SAY$):: GOTO 170
0
```

TELEPHONE DIALLER PROJECT

by Peter Walker & Mike Goddard

In this two part article we intend to demonstrate how one can use the TI99/4A to dial telephone calls. In this first part Peter describes the general principles and software. In part 2 Mike describes typical hardware to achieve this.

**WARNING:** The hardware and software described in these articles is prohibited from connection to the British Telecom network. The information given here may only be exploited on private systems or systems in other countries where regulations may be different. TIUG(UK) neither encourages nor condones violation of the law by its members. These articles are designed to demonstrate the flexibility of the TI99/4A to undertake tasks involving control of external devices with precise timing requirements.

In several recent issues of TI\*MES you may have noticed reference to Interrupt Driven programs. This software also uses this technique to achieve precise pulse timing required on the telephone network. This technique is not described in the Editor Assembler manual, so a little explanation may be required. During the running of a Basic program the computer interrupts the program execution every 1/50th of a second to carry out several background tasks, such as scanning to see if FCTN BREAK or FCTN QUIT have been pressed and to perform certain VDP functions. (In the US the interrupt is every 1/60th of a second - the rate lines up with the TV framing speed). During each interrupt, the machine will also perform any assembler program whose start address is stored at >83C4. During machine code program running, this interrupt arrangement is only activated if the Interrupt Mask is enabled by the LIMI 2 command. You may have come across this command in connection with generating sounds, since sound timing is also associated with the interrupt cycle.

Before discussing the software, it is best to describe the principles on which telephone dialling works. See figure 1. When the telephone handset is lifted, the line is looped (ie closed) by the instrument. Dialling is performed by breaking the loop by a sequence of digit trains. Each digit is represented by breaking the loop the same number of times. For example, digit 3 breaks the loop 3 times. Each break usually lasts 66ms and intervening make periods last 33ms. A complete pulse (make & break) lasts 100ms ie 10 pulses per second. Pulsing is usually performed by a short circuit on the loop. In other countries the pulsing may be 50ms/50ms. Individual digits in the dialling sequence are separated by an inter-digit pause, which needs to be 800ms for older exchanges, though some modern ones need only 300ms. Before dialling starts, it is also necessary to wait until the exchange has responded to the new call and returned dialling tone. Again this varies, but should not in general exceed 2 seconds; it may be much shorter on modern exchanges.

The software for the telephone dialler comprises:

1. An ExBas program which handles the main logic functions and controls the pre-sending and inter-digit pauses which are not so time critical.
2. A machine code routine PULSE which is called by the ExBas program with the digit as input parameter. This program initialises the interrupt driven routine called INT which actually does the pulsing. PULSE waits for the digit pulse to complete before returning control to the ExBas program for the next digit.



3. A machine code interrupt program called INT, which when initialised by PULSE is executed once every 20ms. Digits are pulsed by breaking the loop for 3 cycles (60ms) and making the loop for 2 cycles (40ms). This process is repeated for the number of the digit being pulsed.

4. A machine code program called LOOP which controls the connection of the short circuit pulsing loop to the telephone line.

Figure 2 shows the general hardware principles. All interfacing with the telephone currents is via relays, driven by the cassette port controls CS1 and CS2. (See TI\*MES no 23 for general discussion of cassette port control). CS2 controls the loop connection via program LOOP and CS1 is the pulsing relay controlled by PULSE & INT.

Figure 3 shows a flow chart of the machine code routines. PULSE sets the digit required into Register 4 and sets Register 5 to 5. It initialises the program INT and then loops testing R4 for a zero value. In order that INT is allowed to run, this waiting loop temporarily allows interrupts with LIM1 2. When R4=0 this signals that INT has finished pulsing the digit. INT is disabled and control returns to the ExBas program.

INT is performed every 20ms during digit pulsing. R5 controls the 100ms pulse cycle, while R4 counts down the pulses completed. See Figure 4. When initially called, R5=5 and the loop is opened (CS1 OFF). 60ms later when R5=2 CS1 is switched on again and the loop closed. 40ms later when R5=0 the loop is opened again if a further pulse is required, or if R4=0 the program exits without action and will then be disabled by PULSE which will be waiting for R4=0. INT decrements R5 each time it is run and R4 every time a 100ms cycle is completed.

LOOP is a straitforward program that operates or releases CS2 depending on the input parameter. It is worth remembering that both CS1 and CS2 are both normally on when the TI99/4A is switched on, but they do both pulse off for about 1 second during a hardware reset. Thus care must be taken, using extra manual switches, for example, to avoid interfering with the telephone line when not intended. Alternatively, it is possible to provide a more complex form of control via the Joystick port outputs, but we will not be describing this further here.

The ExBas program can be tailored to your own needs, the one given here demonstrates the general principles. It assumes that the power for the relays and connection to the line are controlled by a separate manual switch. My own version has been improved by the addition of a menu of pre-stored names and numbers.

Some of you might wonder why I have gone to such lengths to provide timings via the interrupt system when it would be possible to do all the make and break timings by delay loops in the PULSE program. Firstly, I found it difficult without the right test gear to create delay loops of the right length - previously published programs quoting time delays by this method turned out to be inaccurate. Secondly, much of the novelty of this article would have been lost. Thirdly, the method shown here is real-time efficient. For example, the PULSE program, rather than just looping on R4 could be doing other background tasks. Equally, the INT program could work with several workspaces in sequence and thus control several external devices "in parallel". This is often how real telephone exchanges work. It would be possible of course to combine the two techniques by providing a

short delay loop of 6.66ms before the MAKE routine, thus simultaneously stretching the break period to its correct value of 66.66ms and shortening the make period to 33.33ms. The need for this depends on how tolerant the pulse receiving device is.

In the next issue of TI\*MES, Mike will describe the hardware arrangements to complement this program.

Some of you may be lucky enough to have telephone exchanges that work with Touchtone multifrequency signalling. Not only is this faster but can be readily generated by the SOUND command in Basic. However, the TI99/4A sound generator actually generates square wave tones, not pure sine wave tones. It is therefore necessary that the touchtone output is filtered to remove the higher frequency harmonics. I can provide anyone interested with further details. If there is enough interest, this could form the subject of a later article.

Peter & Mike

```

100 CALL INIT
110 CALL LOAD("DSK1.PU/03")
120 CALL LINK("LOOP",1)
130 DISPLAY AT(12,1)ERASE AL
L:"TURN ON DIALLER": : "THEN
PRESS ENTER"
140 CALL KEY(3,K,V):: IF V-1
OR K<>13 THEN 140
150 DISPLAY AT(12,1)ERASE AL
L:"INPUT NUMBER"
160 ACCEPT AT(14,1)VALIDATE(
DIGIT)BEEP:P$
170 CALL LINK("LOOP",2)
180 CALL WAIT(1500)
190 FOR J=1 TO LEN(P$)
200 K$=SEG$(P$,J,1)

```

```

210 P=POS("1234567890",K$,1)
220 CALL LINK("PULSE",P)
230 CALL WAIT(800)
240 NEXT J
250 DISPLAY AT(12,1)ERASE AL
L BEEP:"PICKUP PHONE & PRESS
KEY": : "TO RELEASE DIALLER"
260 CALL KEY(3,K,V):: IF V-1
THEN 260
270 CALL LINK("LOOP",1)
280 STOP
290 SUB WAIT(D)
300 CALL SOUND(D,30000,30)
310 CALL SOUND(1,30000,30)
320 SUBEND

```

```

DEF PULSE,LOOP
R11SAV BSS 2
WRKSPC BSS 32
GPLWS EQU >83E0
NUMREF EQU >200C
FAC EQU >834A
STATUS EQU >837C
USINT EQU >83C4 EQUATE FOR USER INTERRUPT

```

\* PULSE PROGRAM - PULSES RELAY ON CS1 PORT

```

*
PULSE MOV R11,@R11SAV SAVE R11 FOR RETURN
LWPI WRKSPC LOAD OUR WORKSPACE
CLR R0 ! PREPARE FOR
LI R1,1 ! NUMREF
BLWP @NUMREF CONVERT INPUT PARAMETER
MOV @FAC,R4 MOVE DIGIT TO R4
ANDI R4,>00FF MASK OFF EXPONENT
LI R1,INT LOAD START ADDRESS IN R1
MOV R1,@USINT ENABLE INTERRUPT PROGRAM
LI R5,5 SET R5=5
LI R12,>002C LOAD R12 FOR CS1 CONTROL
LBL LIM1 2 ENABLE INTERRUPT
LIM1 0 DISABLE INTERRUPT
CI R4,0 R4=0?
JNE LBL LOOP IF NOT
CLR @USINT DISABLE INTERRUPT
CLR R0 !

```



```

OUT3  MOVB R0,@STATUS  ; RESET
      LWPI GPLWS      ; &
      MOV @R11SAV,R11  ; RETURN
      RT              ;
*INTERRUPT PROGRAM RUNS EVERY 20MS WHEN ACTIVATED
INT   LWPI WRKSPC     LOAD OUR WORKSPACE
      CI R5,5         R5=5?
      JEQ BREAK      BREAK IF TRUE
      CI R5,2         R5=2?
      JEQ MAKE       MAKE IF TRUE
      CI R5,0         R5=0?
      JEQ ENDSEQ     JUMP ENDSEQ IF TRUE
OUT1  DEC R5          DECREMENT 20MS COUNTER
OUT2  LWPI GPLWS     RESTORE WORKSPACE
      RT              RETURN
BREAK SBZ 0          RELEASE CS1
      JMP OUT1
MAKE  SBO 0          OPERATE CS1
      JMP OUT1
*SEQUENCE RUN EVERY 100MS
ENDSEQ DEC R4        DECREMENT 100MS COUNTER
      CI R4,0        R4=0? (FINISHED DIGIT?)
      JEQ OUT2
      LI R5,5        SET R5=5 FOR NEXT 100MS CYCLE
      JMP BREAK      JUMP TO BREAK
*LOOP PROGRAM CONTROLS RELAY ON CS2 PORT
LOOP  MOV R11,@R11SAV SAVE R11 FOR RETURN
      LWPI WRKSPC     LOAD WORKSPACE
      LI R12,>002E    SET R12 FOR CS2 CONTROL
      CLR R0          ! PREPARE
      LI R1,1         ! FOR NUMREF
      BLWP @NUMREF    CONVERT INPUT PARAMETER
      MOV @FAC,R1     MOVE PARAMETER TO R1
      ANDI R1,>00FF   MASK OFF EXPONENT
      CI R1,2         R1=2? (OPERATE PARAMETER)
      JEQ RST
      SBZ 0          RELEASE CS2
      JMP OUT3
RST   SBO 0          OPERATE CS2
      JMP OUT3
      END
    
```

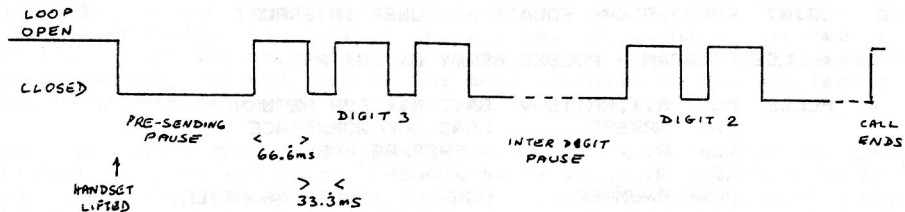


FIGURE 1

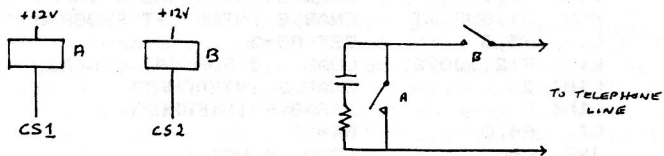


FIGURE 2

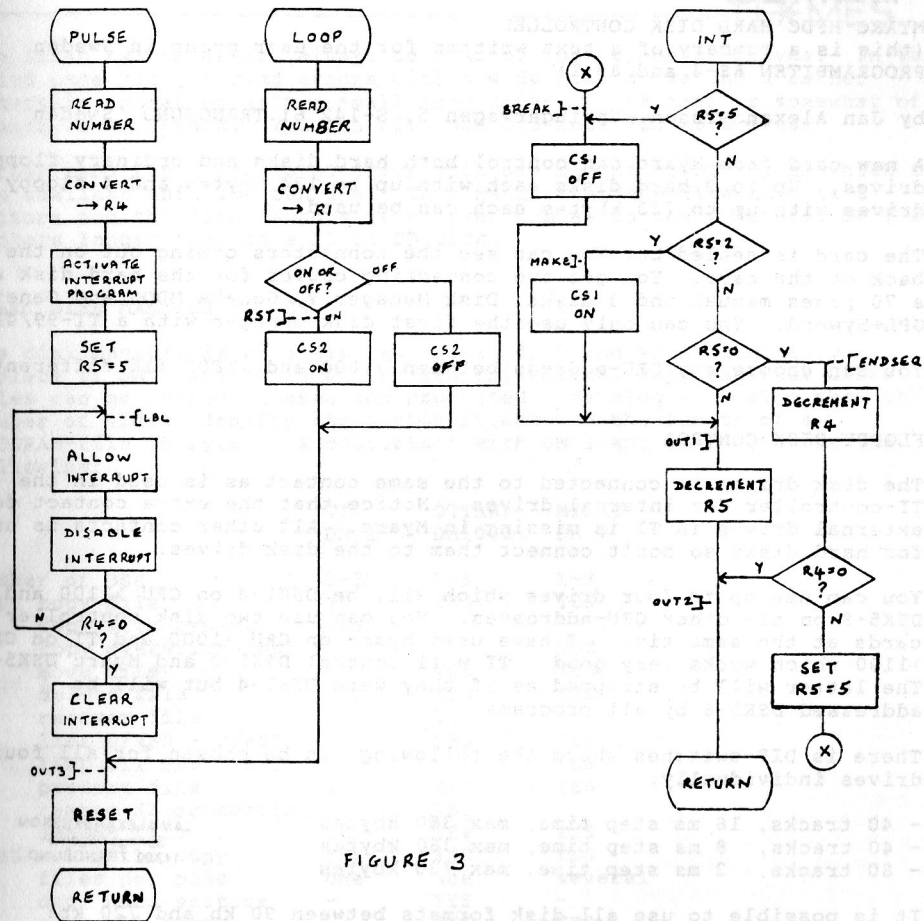


FIGURE 3

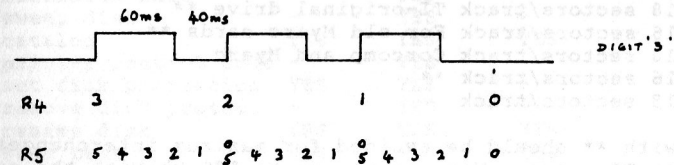


FIGURE 4



# TI\*MES

## MYARC HFDC HARD DISK CONTROLLER

(this is a summary of a text written for the user group in Sweden PROGRAMBITEN 88-4 and 89-1)

by Jan Alexandersson, Springarvagen 5, S-142 61 TRANGSUND, Sweden

A new card from Myarc can control both hard disks and ordinary floppy drives. Up to 3 hard disks each with up to 134 Mbytes and 4 floppy drives with up to 720 kbytes each can be used.

The card is sealed but You can see the connectors coming out on the back of the card. You get two connecting cables for the hard disk and a 70 pages manual and 3 disks: Disk Manager V, Geneve MDOS and Geneve GPL+Myword. You can only use the first disk manager with a TI-99/4A.

You can choose any CRU-address between >1000 and >1F00 (16 different).

## FLOPPY DISK CONTROL

The disk drives is connected to the same contact as is used in the TI-controller for internal drives. Notice that the extra contact for external drives in TI is missing in Myarc. All other contacts is used for hard disks so don't connect them to the disk drives.

You can use up to four drives which will be DSK1-4 on CRU >1100 and DSK5-8 on all other CRU-addresses. You can use two disk controller cards at the same time. I have used Myarc on CRU >1000 and TI on CRU >1100 which works very good. TI will control DSK1-3 and Myarc DSK5-8. The latter will be strapped as if they were DSK1-4 but will be addressed DSK5-8 by all programs.

There is DIP-switches where the following can be chosen for all four drives individually:

- 40 tracks, 16 ms step time, max 360 kbytes
- 40 tracks, 8 ms step time, max 360 kbytes
- 80 tracks, 2 ms step time, max 720 kbytes

JAN ALEXANDERSSON  
Springarvagen 5  
14200 TRANGSUND

It is possible to use all disk formats between 90 kb and 720 kb:

- SS/SD 90 kb 9 sectors/track TI-original drive
- DS/SD 180 kb 9 sectors/track Max with TI-card
- SS/DD 180 kb 18 sectors/track TI-original drive \*\*
- DS/DD 320 kb 16 sectors/track For old Myarc cards \*\*
- DS/DD 360 kb 18 sectors/track Corcomp and Myarc
- DS/QD 640 kb 16 sectors/track \*\*
- DS/QD 720 kb 18 sectors/track

Formats marked with \*\* should be avoided for maximum interchangeability between different TI-users. You can use up to 180 kb with the original TI-drive but it is better to change to double sided drives with 360 kb which also have 180 kb DS/SD, the most common type among TI-99/4A users. You can get a new half height drive for USD 90.

5.25 inch QD (quad density = 80 tracks/side) can read 180 kb SD and 360 kb DD but can only write QD. An 80-tracks drive has a very small width of the read/write head compared to a 40-tracks drive. You can have the following cases:

- 1.tracks from wide head on a fresh disk
- 2.tracks from small head on a fresh disk
- 3.tracks from smallhead written over wide head track

# TI\*MES

The first two cases can always be read by both types of drives. In the third case You get read errors with a wide head drive and a rather uncertain operation with a small head drive if the head is somewhat off center of the track. You can also use 3.5 inch 720 kb drives.

Sector zero for a 720 kb QD drive will use blocks of 512 bytes which is the smallest unit You can use. This means that the file head is 2 sectors and the data sectors is an even number so the file will be 1-2 sectors longer than on a SD or DD disk.

## MYARC DISK MANAGER V 1.29

The disk manager is on a disk and works fine and will completely replace TI DM 2 and DM 1000. It can initialize, copy and test disks. Files can be copied, erased and protected. Catalog will also show the number of sides, density and sectors/track and the length of a PROGRAM-file in bytes. A comparison with DM 2 and DM 1000 shows the following:

	TI DM 2	OTTAWA DM1000	MYARC DM V
Number of DSK	1-3	1-8	1-9
FILE copy file	YES	YES	YES
copy with rename	YES	-	-
copy overwrite test	-	-	YES
move file	-	YES	YES
delete file	YES	YES	YES
recover file	-	YES	-
list D/V80 + D/F80	-	YES	YES
list all D/V + D/F	-	-	YES
protect file	YES	YES	YES
remove XB-protection	-	YES	-
rename file	YES	YES	YES
DISK multifile copy	YES	YES	YES
files per pass	one	one	several
copy used sectors	-	YES	-
copy all sectors	-	YES	YES
init with test	YES	YES	YES
init without test	-	YES	-
multidisk init	-	YES	-
sweep disk	-	YES	-
catalog	YES	YES	YES
printer control code	-	YES	-
set disk protection	YES	YES	-
remove disk protect	-	YES	-
rename disk	YES	YES	YES
TEST read test	YES	-	YES
write test	YES	-	YES

Multifile copy has its own command which is missing in DM 1000 but exists in DM 1000 modified for Funnelweb 4.13. DM V also controls that You don't copy a new file over an old file by the question: File name already exists, Do you want to overwrite (Y/N/All). You can choose Y or N for every old file. If you choose All then all will be copied without any more questions. This is similar to REPLACE in TI-Writer. Several small files is copied in the same pass which is missing in DM 2 and DM 1000. You can also use DM V for Horizon RAM-disk for DSK1-9. This is the first disk manager that can handle DSK9 which is missing in DM 1000.



SUBDIRECTORY ON FLOPPY DISK

A disk can only use 127 files regardless of how it is formatted. It is possible to apart from the ordinary catalog (root) create three subdirectories (DIR) which can hold 127 files each. The disk can then in total contain 508 files because the root can also have 127 files. Each subdirectory has a head which takes one sector. All unused sectors can arbitrarily be used by the three DIR and the root. The use is thus much more flexible than Horizon MENU.

A file in a DIR can be called by DSK1.SUBDIR.FILENAME or DSK.DISKNAME.SUBDIR.FILENAME and a file in the root by DSK1.FILENAME or DSK.DISKNAME.FILENAME. The name of the DIR can be 10 characters long but you should use as short name as possible so it can be within the INPUT-length.

If You have four 90 kb disks then You can easily transfer these to the root and 3 DIR. All four disks have its own catalog so the same file name can exist four times on the disk. Myarc DM V has a special FIND-command that will list all places were a special file name is used.

An other use can be when You write assembler programs and source and object code is saved to different DIR and the PROGRAM-file is saved to the root. A C99-programmer can create DIR for C-code, AL-source code and AL-object code and use the root for PROGRAM-files.

NEW BASIC COMMANDS

You can use OLD, SAVE, OPEN, DELETE and LIST in the usual way but also with DIR like SAVE DSK1.XB.GAME.

There are seven new CALL's which can be used in BASIC command mode and RUN of a program. For Extended Basic you can only use them in command mode.

CALL FILES decides how many open files there could be on the disk drives. The maximum number is 9.

CALL ILR is the same as CALL INIT.

CALL LR("DSK1.OBJECTFILE") is the same as number 3 LOAD and RUN in the Editor/Assembler-module. The command will thus load an object file in DIS/FIX 80-format.

CALL LLR("START") is the same as CALL LINK and will thus start a program as number 4 RUN in the EA-module.

CALL MDM will load DM V from DSK1.

CALL DT will set the clock. You can also do this with OPEN #1:"TIME" and PRINT #1:SEC\$,MIN\$,HR\$,DAY\$,MON\$,YR\$. You can read the clock with INPUT. This is a hardware clock so it will operate correct also with 50 Hz and access to external devices. It has no batteries so it must be set every time You start the computer.

CALL DIR(1) or CALL DIR("DSK1") shows the catalog. CALL DIR("DSK1.SUBDIR") shows the catalog for the subdirectory. A good thing is that the length of the PROGRAM-files is shown in bytes both for Basic and Assembler.

In the same way as for the TI-card you can read the catalog by opening a file with

OPEN #1:"DSK1.",INPUT,RELATIVE,INTERNAL  
where "DSK1." can be replaced by "DSK1.SUBDIR." to get the catalog for a DIR. Notice that the dot shall be used here but not in CALL DIR.

When You read this file You get:

INPUT #1:"DISKS,ZERO,TOTAL,USED

INPUT #1:"FILES,FILETYPE,SECTORS,LENGTH

There is some differences between TI and Myarc. TOTAL=360 and 720 for Myarc when TI has 358 and 718. Myarc shows the length also for PROGRAM-files when TI shows 0 for these. Myarc has a new FILETYPE=6 for subdirectory and FILETYPE=7 for emulate.

A thing that was unknown to me is that files opened with OPEN and DISPLAY has a maximal length of 150 according to the manual, but can handle 156 when I test it. The same is true for the TI-card. You can also open a file with RELATIVE 400 and 400 records is reserved from the start. This will store all records close to each other on the disk which speeds up search of a record. The same thing works both with the Myarc-card and the TI-card. If You open the file a second time with RELATIVE 1000 then Myarc will reserve more space on the disk which the TI-card will not do.

HARD DISK

Up to three hard disks can be used. Each hard disk is connected with an individual address cable and a common data cable. You can use up to 134 Mbytes per hard disk. The best size is 20 Mbytes which is the usual size for most personal computers. The cost in Sweden is about SEK 2000 + VAT. The most common type is from Seagate:

- ST-225 20Mb 65ms 5.25inch
- ST-125 20Mb 35ms 3.5 inch

The hard disk must have a ST506/412 interface but RLL cannot be used. Many 32 Mb drives have RLL so avoid these. If You want a bigger drive then You must choose 40 Mb. You can get a removable 5.25 inch frame to a 3.5 inch drive without any extra cost.

The PE-box cannot give power to the hard disk so You must get an other power supply. The critical voltage is 12 V from which a 5.25 inch needs about 2.5 A and a 3.5 inch needs 2.0 A.

Format of the hard disk takes 4 minutes including test for a 20 Mbytes ST-125 drive. After formatting I got Used 66 sectors Free 78654. With my earlier EPROM H6 I had 2 bad sectors (used 70 free 78650) and with reformat after a week I had no bad sectors (used 68 free 78652). I hope these 2 extra sectors (more than the manual) are no bug. A new thing with EPROM H11 is that it reserves a number of sectors (default 2048) for file heads and directories which will speed up the search and loading in the same way as Mike Dodd's MCOPI for floppy disks. The noise from the hard disk is much less with this new EPROM. The very fast format is done with test at the same time. I am somewhat suspicious about this but today I have no bad sectors so I cannot try it. If You have a hard drive with bad sectors then You can compare the number of bad sectors after format and after a special test program that takes 20 minutes. Myarc says that they use multiple sector I/O in the first case but not with the test routine. It is possible that there is some kind of track-reading in the first case which is not suitable for TEST. The test of the hard drive does a read and write test without destroying data so when You start it You may not break.



small bug in DM V format of hard drive force You to have it loaded from DSK1 even when You have changed reload to DSK5.

Each DIR takes 4 sectors and an usual file is one sector longer than on a floppy disk.

The hard disk have the root and subdirectories in any number of steps below each other. The root and each subdirectory can have 127 files + 114 DIR. These new DIR can also have 127 files + 114 DIR and so on. This means that You can call files by WDS1.GAME.XB.PB.LANDER or WDS.NAME.GAME.XB.PB.LANDER.

The Myarc card also has 32 kbytes RAM (bankswitched). There is always space for 11 open files so CALL FILES decides only the number of files on the floppy disks. The remaining files i.e. usually 11-3=8 can be opened at the same time on the hard disk.

In Micropendium march 1989 there is a program in assembler for parking of a hard disk. Parking means that You position the read/write heads in a place where You have no data.

#### EMULATION OF DSK1, DSK AND DSK1-FILE

Some programs need the files to be on DSK1. You can create a subdirectory DSK1 so that the file can be loaded from hard disk even when it is called by DSK1.CHAR1. All other use of DSK1 is directed to the floppy disk. You can also use subdirectories below DSK1 without WDS1. I have used DSK1.FW.CFG which is good when the input only allow a small number of characters. If You use this DSK1-emulation then You should use the physical drive DSK1 is little as possible. All access to DSK1 goes first to the hard disk and then to the floppy drive. This can slow down some programs considerably which can be seen with Spellcheck or sorting with TI-BASE. Use Your old 90 kb as DSK1 and use Your bigger drives for DSK2-8. Usually DSK1 is only needed for copy protected disks like MG Explorer, MG Diskassembler, MG Diagnostics and Turbo-Pasc'99 (only from TI or Corcomp). All other programs can be placed in the DSK1-emulation (max 127 files, size not important) or if possible with an other path to the hard disk. All file names in the DSK1-emulation cannot be reached on the floppy disk in drive 1.

Programs like Multiplan need files which is called by DSK.TIMP.MPBASE. In this case You create a subdirectory DSK which direct the call to the hard disk.

There is a third emulation called File Emulation for DSK1 (only CRU >1100) which is an exact sector for sector copy of a disk stored under one single file name. This works very good with FORTH which access sectors directly without any files. Several such emulations can be stored simultaneously on the hard disk but only one can be active at the same time. When it is active all calls will go to the hard disk including a sector editor. Also ordinary disks can be stored in this emulation. The search goes first to the DSK1-emulation and if the file is not there then the search goes to the file emulation. There will be no access to the physical drive 1. When this file emulation is active then DSK1 becomes DSK2 and DSK2 becomes DSK3.

#### PROBLEMS

The Myarc HFDC was delivered to me with EPROM H6 and later I sent it back to USA for repair and change to EPROM H11 with DM V 1.29. Myarc

replaced a socket for one chip that prevented the clock from operating. I sent the card to USA as "SMALL PACKET" insured air mail. You shall always investigate the terms for small packet because this is the cheapest way to send cards (Texaments and DIJIT use it but not Myarc).

If You don't use a hard disk then the first access to DSK1 will take 45 seconds before anything happens. With a connected hard disk this problem disappears but You should not buy the HFDC card without a hard disk.

In a letter to me from Myarc they say that some Fujitsu-drives is marginal drives and may not be compatible with Myarc HFDC especially those with stamped steel frames (zinc dye cast is ok). Back-up of hard disk only works to DSK1-3 and not to DSK5-8. CALL MDM only works on CRU >1100 or if You have the MDM5-file in the DSK1-emulation on the hard disk. You must always load MDM5 from DSK1 the first time and then change in SETUP of RELOAD of DM V to your disk number like DSK5 or a path on the hard disk like WDS1.MYARC. It is a good thing to always have several disks ready with different paths so You can get the disk manager if something goes wrong with the DSK1-emulation. MDM5 is sensible to the load path and I have experienced two problems but the usual load with Editor/Assembler or XB DSK1.LOAD is OK:  
 - Directory-Utility-Complete-Catalog can crash the screen because it never waits for input prompt (XB-FW-MDM5 or GK-FW-MDM5).  
 - When DM V is completely loaded then the screen is locked and no key can be used (TW-UTILITY-MDM5).

You cannot set the clock from DM V start menu but if You set it from Basic then it can be used to mark files and directories.

Myarc writes SD-sectors to the disk with deleted data marks F8 instead of FB for a TI-controller. ID/DATA SEPARATORS will be 00.

An other problem comes when You copy disks. When Myarc writes a sector to a disk it will not do a read of the same sector for checking. The problem is the same both with DM 1000 and DM V when Myarc HFDC is used. It is thus very important to verify the disk when You format it. DM V has always verify but DM 1000 has the possibility not to do it. A TI-controller does always a READ after each WRITE of a sector. I find this much more secure. Myarc confirms this in a letter to me and says that MS-DOS works in the same way without a READ after WRITE. Can someone who knows IBM PC confirm this.

After long tests with DM V then I have decided not use Myarc HFDC for DS/SD 180 kb. I have tested it with four different drives TEAC, Fujitsu (2) and Mitsubishi with the same bad result. The comprehensive test takes 10 minutes per loop and I run up to 10 loops which takes more than two hours. With single density SD will the computer always (several on each drive) lock-up after 1-5 loops. With DS/DD 360 kb I have run 10 loops (twice) on all drives without any problems (total 80 loops). DS/QD 720 kb also works perfect. I am not sure if the fault is in Myarc or in the disk drives. Is Myarc too fast or the drives too slow or will 50 Hz mains slow down the TI-99/4A (CALL SOUND and interrupt clock is 20 % too slow).

DS/QD 720 kb floppy drives shows different number of sectors for a file with CALL DIR and DM V but I don't think it is any problem. If You save a very short text from TI-Writer to a QD-disk it will have an extra sector (apart from the problem above) compared to a Save File to a SD- or DD-disk. When this bigger file is copied to SD or DD then the file is still too big but I don't think it is a real problem. I will investigate this more in the future.



I have seen reports about heat problems with other Myarc-cards. Myarc has a sealed card with no cooling of the voltage regulators. One Geneve user in Sweden had a real problem so he has decided to move the card far away from the warm AC/DC-converter (don't use slot 1), remove the card sealing and change the AC-setting to 240 V instead of 220 V. I have also seen concern from Australia about the Myarc 512 kb RAM disk and a possible need for an extra heatsink.

FUTURE EXPANSION

Myarc has prepared the card for tape streamer and 1.44 Mbytes floppy drives. Software for these is not delivered with the card.

A tape streamer is an easy way for back-up of a hard disk. The prize is about SEK 3200 + VAT in Sweden so You may instead want to buy a second hard disk. DM V has routines for back-up of floppy disks but You need a lot of them for a full hard disk. Each file has a back-up flag so You need only to save changed files to the floppies.

A DS/HD 3.5 inch drive can store 1.44 Mbytes and has 80 tracks/side and 36 sectors/track. HD stands for High Density. These drives costs 20 % more than 720 kb 3.5 inch and the floppy disks costs more than the double. Today there is no economical reason to buy 1.44 Mb. The disks will store max 508 files if You use the root and three subdirectories.

REFERENCES

Micropendium:

- mar 88: Myarc ships HFDC
- mar 88: Power supply for hard drive
- apr 88: Organizing your hard disk
- sep 88: Myarc HFDC review
- sep 88: Tips for new hard disk users
- oct 88: Hard disk backups and downloads
- dec 88: Using HFDC emulate files
- mar 89: Parking your hard drive

LETS TELECOMMUNICATE!!

By Peter Walker

A number of our members possess modems, but I wonder how many really feel they make good use of theirs. I would like to start a "Telecoms Special Interest Group" for those members who would like to correspond about Comms issues our try on-line communications eg:

Mailboxing via Prestel

Exchanging program or DV80 files between users  
TE2 and X-Modem error correcting transfers

Members might also exchange information on useful BBSs they have found, or discuss merits of the various terminal emulation programs around, eg TE2, Fast-Term and Telco.

Anyone interested, please write to me (address on cover) or leave a Prestel MBX on 707873778.

```

X X BBBB # 12
X X B B # 13
X BBBB By
X X B B Jim
X X BBBB Swedlow
    
```

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

MORE ON PRINT SEPARATORS

If, in a PRINT USING, you follow the print list with a print separator, your 99/4A will honor it. Consider this:

```

10 FOR I=1 TO 30
20 PRINT USING "## ## ":I,I*I;
30 NEXT I
    
```

Instead of one column of numbers and their squares (which you would have without the semi-colon after the I\*I) you will get four right justified columns on your screen. Also works with printers and with DISPLAY USING.

TE II ABILITY (TE2 is a module)

With a program in memory and your Speech Synthesizer attached, enter LIST "SPEECH" and see what happens. Strange! (Source: Regina in COMPUTE)

LINE NUMBER TABLE

What, you ask, is a line number table? Glad you asked! When your TI stores a program in memory or on disk, it is broken into two parts: the line number table and the line contents. In this program:

```

100 ! TI
110 FOR I=1 TO 10
120 PRINT I
130 NEXT I :: CALL CLEAR
    
```

The line number table looks like this:

HEX	DECIMAL
00B2 37B5	130 14261
0078 37C2	120 14274
006E 37C6	110 14278
0064 37D3	100 14291

Note that the line numbers are in descending order. The second number is the memory location of the line contents (this is only an example). For example, the contents of line 100 start at >37D3. The > symbol means that the number is in hex or hexadecimal or base 16.

When you RUN a program your 4A, during pre-scan, reserves memory for variables and other things. That's why you get different responses to SIZE before and after RUNNING a program.

While your computer always keeps the line numbers in descending order, it shifts the memory locations of the line contents around without regard to order as you edit/modify your program.

If you save a program to disk in MERGE format, enter NEW and then MERGE your program back into memory, your line number table should be returned to order. This may or may not have any real advantage, however.



What use is this? It depends. If you want to go looking in memory and learn how TI stores and executes a program, this is the first step.

If not? Well, now you know that each line number takes four bytes of memory (two for the number and two for the memory address). A line separator (:) takes only one byte. Conclusion: multi-line statements save memory!

FILE\*PRINT

Many files are stored in DISPLAY VARIABLE 80 format: TI writer files, Editor Assembler source files, text files, documentation files, etc. This month's program reads your disk for DV80 files and displays a menu on your screen.

You can then print up to 99 copies of any file with a single keystroke. Further, if your printer is GEMINI/EPSON compatible, you can activate double strike and/or emphasized print.

If you get strange error messages when you first run this program, change the ON ERROR statements in lines 170 and 270 (twice) to ON ERROR STOP and change the :@- in line 150 to ;. After you have finished debugging, restore these commands.

Enjoy!

```

100 : FILE:PRINT
110 : VERSION XB,1.0
120 : 26 MAY 85
130 : By Jim Swedlow
140 :
150 N:=1 : DIM N$(24),S$(1
: E$(1),D$(1),T$(1) : GOTO 1
60 : CALL CLEAR : CALL KEY
: A,B,D,E,H,T,A$,P$ : IF P$
- : S$(0)="OFF" : S$(0)=P$
: P$=CHR$(27) : E$(0)=P$
" : : D$(0)
" : : E$(0)=P$&"E" : : D$(0)
" : : D$(0)=P$&"G" : : T
$)=CHR$(12) : P$="P10"
170 DISPLAY AT(10),ERASE AL
L BEEP:"FILE:PRINT" : :
: Reading Disk Directory" :
: ON ERROR 330 : H=0 : DPE
N#:"DSK1,"INPUT,RELATIVE
INTERNAL : INPUT #A$,B$,A,B
: B
180 INPUT #A$,A$,B$,B : IF
A<>" " THEN 180 ELSE H=H+@ :
B<80 THEN 180 ELSE H=A$ : GO
190 CLOSE #@ : ON ERROR STO
10 180
IF H<25 THEN N$(H)=A$ : GO
270 DISPLAY AT(20,@):Printf
ng "N$(B-64) : : : : ON
ERROR 340 : OPEN #2:P$ :
PRINT #2:E$(E);D$(D) : ON ER
ROR 330 : OPEN #@:"DSK1,"RN
$(B-64) : : FOR A=@ TO N : : RE
STORE #@
280 IF EOF(@)=0 THEN INPUT
#1:A$ : PRINT #2:A$ : : GOTO

```

```

200 DISPLAY AT(5,@) : GOSUB
300 : B=14 : : FOR A=@ TO H
: B=15-B : : DISPLAY AT(INT
(A+@/2)+6,B):CHR$(64+A) : "
: N$(A) : NEXT A
210 DISPLAY AT(20,@):"Press
sized <2> # Copies<3> TO A
< > to change:" : : <1> Empha
t End <4> Dbl Strk<5> Print
er
220 GOSUB 300
230 CALL KEY(3,B,A) : IF A@
THEN 230 ELSE IF B>64 AND B
<65+H THEN 270 ELSE IF B=54
THEN CALL CLEAR : : STOP
240 IF B=49 THEN E=@-E : : GO
TO 220 ELSE IF B=52 THEN D=@
-D : : GOTO 220 ELSE IF B=51
THEN T=@-1 : : GOTO 220
250 IF B=50 THEN ACCEPT AT(3
:26)SIZE(-2)VALIDATE(DIGIT)B
E:P;N : : N=MAX(N,@) : : T=1-(N
>@) : : T=MIN(T,@) : : GOTO 220
260 IF B=53 THEN ACCEPT AT(5
:9)SIZE(-20):P$ : : IF P$=" "
THEN 260 ELSE 210 ELSE 230
270 DISPLAY AT(20,@):Printf
ng "N$(B-64) : : : : ON
ERROR 340 : OPEN #2:P$ :
PRINT #2:E$(E);D$(D) : ON ER
ROR 330 : OPEN #@:"DSK1,"RN
$(B-64) : : FOR A=@ TO N : : RE
STORE #@
280 IF EOF(@)=0 THEN INPUT
#1:A$ : PRINT #2:A$ : : GOTO
280

```

ON PROGRAM FILES

When you catalog a disk, there are five types of files: Program, Display Fixed, Display Variable, Internal Fixed and Internal Variable.

A program file varies from the others as they are a collection of individual records (data) while a program file is a "memory image".

Memory image means essentially that, when writing to disk, your 4A takes what is in RAM and dumps it to disk. In other words, it is a copy or image of what there are at least three types of program files: Basic and XB programs, Assembly language files saved in program format and data files produced by some modules (Personal Record Keeping, Adventure, etc).

It is only when you attempt to load and execute that your computer notes the difference (and it often locks up if you load a program file that does not match the application).

Program files (be they programs or data) load significantly faster than other files. Why? Well, think of a bag of marbles. When in memory image, loading the file into RAM is a matter of pouring the bag from disk to CPU. In any other format, however, you take the marbles one by one from disk to memory.

Dumping the bag is simply faster.

Finally, Assembly language programs in mirror image format can be loaded from the Editor/Assembler module or from TI WRITER's option 3. In both cases, the module will auto boot a program entitled DSK1.UUTIL.

```

290 PRINT #2:T$(T) : NEXT A
: : PRINT #2:E$(0);D$(0) : CL
OSE #2 : CLOSE #@ : GOTO 2
: : " : : found in drive 1
: : " : : Insert disk in driv
10 : : " : : ON ERROR 350 : CLOS
e 1" : : ON ERROR 350 : CLOS
E #@ : : ON ERROR 350 : CLOS
E #2 : : GOSUB 310 : : GOTO 17
0

```

```

350 END
350 RETURN NEXT
53 : GOTO 260

```

```

330 DISPLAY AT(3,@):BEEP : :
340 ON ERROR 350 : : CLOSE #2
: : DISPLAY AT(20,@):BEEP:"Pr
inter name is invalid" : : B=
Press any key to continue"
320 CALL KEY(3,A,B) : IF B@
THEN 320 ELSE RETURN

```