

NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 12, Number 2

March, 1993

Registered by Australia Post - Publication No. NBH5933



See you at the next meeting
Dick Warburton.

Sydney, New South Wales, Australia

\$3

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

TiSHUG News Digest

March 1993

All correspondence to:
C/o 3 Storey St.
Ryde 2112 Australia

TiSHUG News Digest

ISSN 0819-1984

INDEX

The Board

Co-ordinator

Dick Warburton (02) 918 8132

Secretary

Russell Welham (043) 92 4000

Treasurer

Cyril Bohlsen (02) 839 5847

Directors

Percy Harrison (02) 808 3181

Sub-committees

News Digest Editor

Bob Relyea (048) 57 1253

BBS Sysop

Ross Mudie (02) 458 2122

BBS telephone number (02) 458 4606

Merchandising

Percy Harrison (02) 808 3181

Publications Library

Russell Welham (043) 92 4000

Software Library

Larry Saunders (02) 844 7977

Technical Co-ordinator

Geoff Trott (042) 28 6629

Regional Group Contacts

Central Coast

Russell Welham (043) 92 4000

Coffs Harbour

Kevin Cox (088) 53 2848

Glebe

Mike Slattery (02) 692 8182

Hunter Valley

Geoff Phillips (049) 42 8178

Mawarra

Geoff Trott (042) 29 8829

Liverpool

Larry Saunders (02) 844 7377

Northern Suburbs

Dennis Norman (02) 452 3920

Sutherland

Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues \$35.00
Associate membership \$10.00
Overseas Airmail Dues A\$85.00
Overseas Surface Dues A\$50.00

TiSHUG Sydney Meeting

The March Meeting will start at
2.00 pm on the 6th March
at Ryde Infants School,
Tucker Street, Ryde.

Printed by
Kwik Kopy Parramatta

Title	Description	Author	Page No
Assembly Bit Reversal Routine	Software Hints	Ross Mudie	17
Co-ordinators Report	General Interest	Dick Warburton	2
Demonstration of Metex Data Logging	Notice	Ross Mudie	21
Editors Comment	General Interest	Bob Relyea	1
For Sale	Notice	Geoff Trott	23
Gif Files on the BBS	Notice	Ross Mudie	22
Implanting D/F 128 Files	Hints	Ross Mudie	7
Interfacing a Metex Digital Multimeter	Software Hint	Ross Mudie	19
Learn to Know Your TI	General Interest	Percy Harrison	9
Making a Recording Voltmeter	Software Hint	Ross Mudie	11
Nostalgia Time	General Interest	Geoff Trott	10
Regional Group Reports	Notice	Jim Swedlow	23
Techo Time	General Interest	Geoff Trott	5
The Art of Trouble Shooting	Hints	Ben V. Tokach	22
The Communicators	General Interest	Ross Mudie	3
TiSHUG Shop	Club Notice	Percy Harrison	18
TiSHUG Software December	Software	Larry Saunders	14
February	Software	Larry Saunders	15
March	Software	Larry Saunders	16
Treasurer's Report	General Interest	Cyril Bohlsen	16
Tutorial Weekend Change of Date	Notice	Ross Mudie	21
Word Processing Part 3	Software Hints	Col Christensen	12

Notice

Due to lack of interest the Console Repair Night held on the Friday following the General Meetings will be discontinued for the present time.

Editor's Comments

by Bob Relyea

Just a few words this month to say a 'thank you' for all the local articles that have come in. It might be difficult to fit it all in this issue, even with extra pages but do not let that discourage you- keep them coming in. Most of the local gear is still from the same faithful members but we are still looking for a wider range of members to contribute. I apologise to Ross for not getting his gear off the BBS in time to make the first issue of the TND. We will try to make up for it this month, Ross.

CO-ORDINATOR'S REPORT

Sex and the TI-994A.

Is there a connection? For a start most TI Computer users are male. Do their spouses share their interest? Unfortunately, rarely. Is that because TI users tend to avoid going to bed and spend their hours either burning the midnight oil in front of their trusty TI, or attending a computer function in some far off suburb, and getting home with the milkman? For that matter, why do they choose to have fun only with other males? Is the TI a substitute for sex? And what is the neglected spouse's substitute?

Some club members appear to love their TI more than other members of the family? I hate to admit it, but I have actually heard TI users talk to their machines? I've even had a word or two in that direction myself. Stern words. Masterful words. Authoritive. Expletives. The trouble is, the TI, being the advanced little monster that he is, can even manage to talk back. So... Do computer users actually fall in love with their machines? Substitute their computer for relationships with people? Uh, oh! Never.

Never? Whatever the motivation, many users become hooked. The computer takes over. The user needs a fix every day. No computer, and withdrawal symptoms set in. Irritability. Onset of "temperament". Until resumption of that oh so necessary daily stimulation.

Naturally it's much more enjoyable to use a computer than to mow the lawn. Or fix the fence. Or drive the kids to sport. People even accuse dedicated computer users of being less fit than the average person. Are we really heavier? Well, we certainly sit down more. But eat more? Only a fool would eat and drink whilst at your machine. Or would they? It's amazing how hunger, and time become irrelevant. In fact, we're never late for meals, simply because mealtimes don't exist. Microwaves become necessities. I mean, who could desert their system when success is just around the corner.

Is the TI a sex symbol? Freud I'm sure would have found a connection, but then he seemed to believe most human behaviour was driven by an underlying sexual appetite. Certainly a few users enjoy their "adult" graphics or games, and it's true that computer usage may completely submerge strong drives like sex or hunger. Become an end in itself?

Actually, the TI becomes one of the family. Like family, it can determine not only our interests, but our friendships. In discovering the TI, I opened up a whole world of new friends - and a total lack of boredom. The Texas provides a constant challenge, an ever present stimulation. Not to mention a perpetual mastering of new concepts and skills. Yes. Friendships increase. Communication skills improve. Brain cells decay more slowly.

Have you ever noticed how dull and boring people become when they lack absorbing interests. In brain cells, like muscles, lack of exercise produces a reduction in efficiency. With a mind tool like a computer usage is only limited by imagination, intelligence, and practice.

I hate to admit it, but... the TI is not the only computer. Funny thing, other computer users can be just as motivated, even obsessive about their system.

For many of us, the TI was our first computer. As available technology escalates at a mind boggling pace, and software becomes faster, with more powerful features, better user friendly interfaces, and truly beautiful supporting graphics, goals change. Some TI users have been dismayed - then excited - to find their work or personal requirements are occasionally even better met by using a more modern updated machine. Not necessarily an inherently better machine, but one better supported by peripheral hardware (did you order an 80 column card?), and software programmers because of the infinitely larger market.

The TI remains as a first love, a valued love. But not an only love. The TI opened the door to infinite treasure.

At least two club members futures are assured, primarily because they began with a TI at an early age. All living things grow and change. Without growth we die. A new era is dawning for computer users. Modern machines are far more powerful than the mainframes of a decade ago. Better machines mean the ability to create fantastic graphics, compose music, write creatively, live in a world of virtual reality, have access to immense data bases in every area of knowledge, at the touch of a key or mouse.

Libraries will change, work will change, patterns of living will be drastically altered in almost every conceivable way. Business. Banking. Buying. Even medicine. Certainly research and development.

Whatever the changes, TI users will be better prepared than most to adjust and cope with the degree of change. Why? Because they have grown up with this new technology from its shaky uncertain beginning. Unsupported from outside, they have learnt to be independent and self reliant, perhaps more than any other computer group.

To cater for the natural evolutionary changes which are occurring within the club, and as more and more members become familiar with other machines and technology, we will set up an interest group for club members to explore their expanding interests and knowledge. Interested members who attend club meetings, are invited to meet at 12 PM each month to extend their interest.

Has sex got as little to do with the TI as the last few paragraphs? Perhaps the answers would be as individual as our members. One thing is clear. Computer usage can be as motivating and as interesting as sex for a number of users. If you had to make a choice between ongoing sex and using your computer, what would you choose? I wonder?

See you at the next meeting

Dick Warburton.

CONTINUED FROM PAGE 8

* The following memory assignments are placed outside
* implant area to minimise memory usage in the implanted
* area.

SAVRTN EQU >3F00 2 bytes from >3F00 to >3F01
WS EQU >3F02 >20 32 bytes from >3F02 to >3F21
BUFFER EQU >3F22 >51 82 bytes from >3F22 to >3F72

GIFUNS TEXT 'GIFUNS' This is the DEF table entry.
DATA START The x/b CALL LOAD(8196,255,152)
* redefines the location of the DEF table.

START MOV R11,@SAVRTN
LWPI WS

S @D128,@STADDR Next address to get a string

CLR RO Element zero in variable A\$
LI R1,1 Argument 1 in LINK
LI R2,BUFFER Address of BUFFER in LOW RAM
MOVB @D128+1,@BUFFER Specify length for STRASG
MOV @D128,R5 Loop counter control value
MOV @STADDR,R3 Where to get the information
LI R4,BUFFER+1 Where to put it
LOOP1 MOVB *R3+,*R4+ Move it a byte at a time
DEC R5 Finished yet?
JNE LOOP1 If no, go and do the next one
BLWP @STRASG Send the string to X/B

CLR RO }
MOVB RO,@>837C } Text book method of return
LWPI >83E0 } to the calling program
MOV @SAVRTN,R11 }
B *R11 }

D128 DATA 128 Data assignment of value of 128
STADDR DATA >FF98 Value used to commence calculation
* of address of strings. Location is used subsequently
* to remember where in the memory block to continue on
* additional fetches.

END

THE COMMUNICATORS
 PROGRAMS AND FILES ON TEXPAC BBS UNTIL END MARCH 1993
 by Ross Mudie, SYSOP

The number of people using the TEXPAC BBS is falling and some of those who are using the system don't appear to be looking at the available information. In the days when I had more spare time I used to prepare Program and File Information files which described what each file was about. These days I am having trouble finding sufficient time to make a different selection of files and programs each two months. It would be nice if some one could help with this job! The reasons that I have prepared this file at the moment is that I am on holiday (with the computer set up at the holiday house) and I am not staggering home from work in the evenings too mentally exhausted or sick and tired of looking at a computer screen to even want to look at a computer.

When there is no file describing the contents of each file then users don't have to read the whole file if they don't want to. To exit a file early, just press the letter E, once, and the BBS should Exit the file being listed once the BBS modem buffer is emptied. This can take as much as 255 characters or about 6 screen lines, before the Exit actually occurs at the 300 or 1200 baud data rates. To pause a file use CTRL S, the same buffer emptying of the BBS modem will occur before the listing actually pauses. Pressing any other key will restart the listing.

The workload on the SYSOP, that's me, can be reduced if members are prepared to be Sub-Editors. Each Sub-Editor is given one or more Sub-Editor file names and can place information directly on the BBS for others to read. The Sub-Editor can change the information on a suggested 2 monthly basis or more quickly if required. All that is then needed is the members to log on a fairly regular basis to read the information. The top line of each Sub-Editor file is date and time stamped by the BBS when it is uploaded.

The system is also a good way to get information to the TND Editor, provided that the files are not too big and are text only, rather than being accompanied by drawings or photographs, etc.

The following lists detail the contents of the NEWS and PROGRAM menus on the BBS until at least the end of March 1993. At the time of writing I am expecting to be overseas for work from immediately after the February TISHUG meeting for about 2 weeks and on my return I will be involved with some organisation for the Central Coast Amateur Radio Field Day (Wyong race course, 28th Feb.) and probably a 2 week build up of work.

The files are in basically 2 parts. The first part is NEW or TOPICAL or LOCAL information, chiefly from local authors. The second part is the older reference type information which is being repeated on the BBS. This information is still considered relevant and useful. It is quite amazing what I re-discover when looking through the older files.

The downloadable programs are also listed. Programs only be downloaded by a TI99/4A, files can be read by any computer.

LISTING OF TEXPAC NEWS MENU JANUARY - FEBRUARY 1993

FILE
 Author

NEWS, PART 1 NEW OR TOPICAL INFORMATION

ALL
 Anyone The ALL file contains messages from any user to everyone. Its like a notice board with adds, greetings, the occasional plea for help with computer problems, etc. Items are first sent to the ALL MAIL file & later transferred & cleaned up (if needed) by SYSOP. Mail in the ALL MAIL file is read from Main Menu 4.

CO-ORD
 R. Warburton
 Sub-Ed When written (13/1/93) this file contained the Coordinators report that was too late for the December 1992 TND. With a bit of luck Dick may have uploaded a new file, or it may still be a bit of almost unseen history!

GIF
 R. Mudie This is an explanation of whats going on with implanted GIF files on the BBS.

GIF-PPRO
 A. Ruggeri Information about what is planned concerning articles on conversion from GIF to PAGE PRO.

GIF-RVIEW
 A. Ruggeri Features, techniques and good reference information about GIF, (Graphics Interchange Format and the program GIF MANIA which allows these files to be viewed on the TI99/4A.

GIF_IMPL1
 R. Mudie Techniques used to implant GIF files inside Extended BASIC programs for download via the BBS. Single part files, includes assembly source file listings.

GIF_IMPL2
 R. Mudie Extension of the article on single part files to cater for large files which must be split into a number of smaller files to produce manageable size GIF file implanted programs.

METEX_I/F
 R. Mudie How to interface between a multimeter which has a serial data output and the TI99/4A computer. Includes a discussion, electrical interconnection details, Extended BASIC and commented linked assembly language source file listing.

TUT_WKEND
 R. Mudie Ideas for a tutorial weekend for Extended BASIC beginners.

BIT&BITE1
 L.Saunders
 Sub-Ed TISHUG Software releases for December 1992 and February 1993.

BIT&BITE2
 L.Saunders
 Sub-Ed Running of the program POLE POSITION, includes strategy tips for the program.

BIT&BITE3
 L.Saunders
 Sub-Ed Rules for XORKLE.

BIT_REVSE
 R. Mudie Assembly language hints, a routine to reverse the bits in an 8 bit byte or a 16 bit word.

GAMES
 R. Brown
 Sub-Ed The start of SERIES 3 of GAMES INFORMATION from a master of games Robert Brown. This article gives solutions to the Infocom adventure SUSPECT.

LIVERPOOL
 L.Saunders
 Sub-Ed Information giving dates, topics and location of Liverpool Regional meetings.

The BBS provides facilities for all Regional Groups to post their own meeting notices and reports. Only Liverpool is currently using this facility.

STARTING2
 R. Mudie The second in a proposed series to help beginners and those having problems with TI Extended BASIC. The third in this series of articles has been commenced, it this the sort of information that members want?

NEWS, PART 2, REPEATED REFERENCE TYPE INFORMATION

1_LINERS1 Four handy DEF routines for Extended BASIC
T.Atkinson programmers. Rounding numbers to specified
Canada decimal places; Convert natural logs to any
specified base; Using MAX and MIN to find the
highest, lowest and middle numbers of a set
of 3 numbers; Conversion of Joystick results
to integers for faster processing in games
programs.

1_LINERS2 Hexadecimal (base 16) to Decimal conversion;
SubFile99 DEFs for averaging & Mark-up percentage.
USA

ALPHALOCK Details of the single diode modification for
J.Paine the TI99/4A keyboard which overcome the
AUSTRALIA conflict between the up direction of the
Joystick and the Alphalock key.

BEGIN_TIP Tips from the Tiger Cub on EDITing, use of
J.Peterson LET, use of the colon print separator in X/B
USA Dummy DATA statements and RESTORE, changing
the colour of the cursor, character redefin-
ition and disabling QUIT.

BIT_MAP Commented Assembly source file listing about
M.McCormick about Sprites and Text.
USA

BIT_PRINT How to print BIT IMAGE Graphics on Dot
Tom Kennedy Matrix printers from Extended BASIC.
USA

BRANCHING Extended BASIC hints on efficient programming
TEXNET BBS including branching on two variables.
USA

CHAR_CODE A Character Merge utility for use when typing
B. Traver lots of CALL CHARS to eliminate repedative
USA typing.

DEBUG1 Help with debugging assembly language
M.McCormick programs with the SUPER BUGGER program.
USA

DSK_FIXER How to fix disks when software corruption
N.Shah & has occurred. Hints in using the DISK FIXER
M. Ballman program.
USA

DSK_TALK1 Two articles which help to understand how
DSK_TALK2 information is stored on floppy disks. These
SF99 articles supplement the article on the DISK
USA BBS FIXER.

JOYST_ASY Commented assembly routines to assist in
M.McCormick understanding how to access the JOYSTICKS
USA in assembly programming.

LOAD_INT A compilation of articles and programming
R.Binkerd details in assembly and Extended BASIC on the
USA subject and application of LOAD INTERRUPT.

MEM_MAP Memory usage details for CRU, VDP RAM,
R.Coffee Console GROM, VDP RAM (X/B). Essential for
USA programming in Assembly or Linked Assembly
environments.

PRT_USING Hints and HELP on how to use PRINT USING with
TEXNET BBS your printer.
USA

RAMADDRESS A very handy look up table of the use of
R. Mudie addresses in the SYSTEM RAM under Extended
AUSTRALIA BASIC. Includes decimal and hexadecimal
address values, EQUates for assembly utility
programs, data values for GPLLNK (specially
loaded for x/b), XMLLNK in both environments
(which are different) and some pointers.
Team this up with the MEM_MAP file.

RESTART ReUSING already loaded assembly programs from
J.Spiegel Extended BASIC without the need to reload
USA after QUIT has been used.

S&I_CODES An explanation of how Extended BASIC and
TEXNET BBS BASIC store their programs and the codes used
USA in this storage.

SAV_RECAL A method of implanting assembly inside
T.Kaplan Extended BASIC programs. Uses access via the
USA User Interrupt Service Routine.

SPEECH A discussion and program examples on use of
R.Albright unlimited vocabulary speech with Terminal
USA Emulator 2 or the Extended BASIC disk loaded
speech utility.

TI_ALARM Using a TI99/4A console as a Burgular Alarm.
Unknown
USA

TI_W_PRNT A Second look at TI Writer Printer Character
H. Rich Sets.

XB MORSE/S Commented source file on the assembly part of
R. Mudie of a program to generate precision Morse Code
AUSTRALIA from the TI99/4A keyboard. This is a good
tutorial on key board scanning and time share
techniques.

PROGRAM INFORMATION

The following details the selection of programs on
TEXPAC BBS until at least the end of February 1993. The
majority of these Extended BASIC programs require the
32K Memory Expansion and all from GIF MANIA to the end
of the list need a disk system.

Program	System	Comments
4A/FLYER	XB32K	Flight Simulator.
ACE	XB32K	Converts assembly to X/B CALL LOADS.
ARCTURS1XB ARCTURS2XB ARCTURS3XB	XB32K	Space, shooting game, interesting graphics.
ASTIROADS	XB32K	Shooting space game.
BANDIT	XB32K	Maze, pick up treasures.
BOXING	XB32K	2 player game in the boxing ring, uses joysticks.
BREAKTHRU	XB32K	Deflect the bouncing ball and stop it going out of the bottom of screen.
BUCK2RG1XB BUCK2RG2XB	XB32K	Flying and shooting with speech, an Speech arcade game.
BULLRUN	XB	Strategy war game.
BURGER1XB BURGER2XB	XB32K	Help the cook to climb the ladders to make the hamburgers, watch out for the nasties.
CARWARS	XB32K	Try to beat the maniac in the computer
CONGO1XB CONGO2XB CONGO3XB	XB32K	Help the little man thru dangerous territories, watch out for falling boulders and river crossings.
DONKEY1XB DONKEY2XB	XB32K	Donkey Kong, arcade game.
EDITOR-AID	XB	Edits & manipulates XB programs in MERGE format.

CONTINUED ON PAGE 6

Fractured Files

The story I will be telling here relates to a bug in the DSR routines of the Myarc Hard and Floppy Disk Controller Card. I realise that this is not of much interest to most of you but you may be interested to find out how the TI99/4A operating system stores files on disks. In particular, I will be discussing how the operating system manages to keep track of fractured files, which are files which do not occupy consecutive sectors on a disk.

Fractured files can be caused by many events which happen to files on disks. If files are stored one after the other and never deleted or changed in any way, there would be no fractured files on a disk (except possibly one). When all the files on a disk are copied onto a blank disk with no bad sectors using a file copier file by file, all files will be un-fractured (except possibly one). I will give an example of how fractured files occur.

First format a disk, which will leave it with 2 sectors used and the rest full of >E5 bytes. The used sectors are the first sector (0), for the Volume Information Block and sector 1 for the directory. Each sector contains 256 bytes and in sector 0 these bytes contain information about the disk itself along with the bit map of which sectors are free (or used). In this case the bit map will show that sectors 0 and 1 are used and all others are free. The directory sector (1) contains up to 127 entries (two byte numbers) followed by two bytes of zeros. These entries are the sector numbers for the File Information Blocks for each file on the disk, in alphabetical order of file name. If the disk contains no files, this sector will be all zeros.

You can examine the contents of sectors on a disk using DiskReview provided with the latest Funnelweb system. Using an editor, type in 4 lines of characters, each 60 characters long. This will fill up a sector on disk with 240 characters plus a few extras which we need not worry about here. If you save this as FILE1 on the freshly formatted disk, you will find that sectors 0, 1, 2 and >22 are now used. Sector 0 has not changed much, just a couple of bytes in the bit map. Sector 1 now contains >0002 as the first two bytes. Sector 2 has the FIB for FILE1 which tells us that the file is 1 sector long and the sector used is >22. Sector >22 contains the data.

Now go back into the editor and create another file of 4 lines with 60 characters in each line using different characters from FILE1 and save this as FILE2. Now when you look at the disk you will find that the bit map has changed a bit, sector 1 has >0002 >0003 in it with sector 2 as before and sector 3 containing the FIB for FILE2, which tells us that FILE2 is one sector long and uses sector >23 for the data.

If you now go back into the editor, read back in FILE1 and make it twice as big by copying the 4 lines, when you go to save the file it now requires two sectors for the data. It could do that by using sectors >24 and >25 and marking sector >22 as unused. Some operating systems do this and so they always have their files in consecutive sectors but they end up with a lot of space on the disk which is not usable. The TI99/4A operating system re-uses the original sector and looks for the next available sector after >22 to put the rest of the file. In this case it would use sectors >22 and >24 for the data as sector >23 is in use. The FIB would still be in sector 2.

If you repeat the operation on FILE2 of doubling its size and then saving it back, it will use sectors >23 and >25. By this means we have created two fractured files. If we carry on in this way, working on one file at a time, we end up with two files as fractured as it is possible to get. There is a limit to the number of fractures that a file can have (76) before it runs out of room to store the fractures in the FIB.

I hope that explains why the operating system uses fractured files and how they arise. There is nothing wrong with fractured files (normally) and they are just a bit slower to access as they may be spread all over the disk. However, with the bug in the HFDDC floppy disk routines, fractured files seem to cause problems of data overwriting of one file by another. I did not know the exact nature of the problem, only that if I avoided fractured files all worked perfectly. I hasten to add that there is no sign of this problem when using the hard disk but it is there when a floppy disk is emulated on the hard disk. I must also say that I am using version 12 of the DSR, which I have recently discovered is considered to contain too many problems to be usable. I have found that it is quite good except for this one problem.

In order to find out exactly what was happening, I decided to create fractured files exactly as described above and I found that the problems arose after the first fracture. I then followed a slightly different procedure to find out more information and I will now give you the information that I found which explains exactly what the bug is.

A freshly formatted disk (I happened to use 40 track, single sided, single density) has a bit map in sector 0 (starting at byte >38) and taking up 360/8 bytes (one bit per sector) which means it goes from byte >38 to byte >64 inclusive. For a blank disk with no sectors used, all these bytes would contain >00. Each sector in use has a 1 in the corresponding bit position so that with sector 0 and 1 in use, byte >38 will contain >03. The bits get used from the least significant bit in each byte so that sector 0 uses the least significant bit of byte >38. I will not worry about the rest of sector 0, other than the bit map, as it does not come into the scene for this problem. I will give you the first 10 bytes of the bit map each time, as all the rest will be zero. Also, all numbers will be hexadecimal, without the ">" to make it easier. So after the disk has been formatted the bit map will be:

```
03 00 00 00 00 00 00 00 00 00
```

Sector 1 is all zeros at this time and all other sectors contain >E5. After the file FILE1 is stored on the disk as a one sector file, the bit map is:

```
07 00 00 00 04 00 00 00 00 00
```

Sector 2 is now in use to hold the directory entry (FIB) of the file and sector >22 is in use containing the data of the file. The data of interest at this time in the FIB sector (2) starts in byte >1C. Starting here, each 3 bytes from this point on contain the information about which sectors of the disk are used to store the data for this file. At this point in time, these first 3 bytes are 22 00 00, and all remaining bytes of the sector are also zero. This tells us that the first sector used is >022 and the last sector used in this fragment is >022 + >000 = >22 or only one sector is used. The actual number of sectors is also stored in another place in this FIB sector but we will not worry with that either. Also, the number of sectors used by a file that is reported by disk managers includes the FIB sector so it is one more than reported here. The information about the fragments of the file is rather tricky until you get used to it. Let me try to explain it now.

IMPLANTING DISPLAY/FIXED 128 FILES IN EXTENDED BASIC
by Ross Mudie

1. INTRODUCTION

Since the advent of good quality scanned images in GIF format for the TI99/4A, there has been a desire to be able to download GIF files via the TEXPAC Bulletin Board System. The BBS does not have XMODEM which would allow D/F 128 files to be downloaded. A method of allowing unsuitable files to be downloaded is to implant the required file inside an Extended BASIC program which can be downloaded via the existing facilities of the BBS. Inside the special program, as well as the implanted file, there is an "unloader" program. To unload the D/F 128 file, the "Extended BASIC" program is simply run and it writes a D/F 128 file to disk.

The following is a description of the process used to achieve the implant. In time the process may be improved, but here is my current technique for single part files. When the file is bigger than 44 sectors it is necessary to split the file into multiple parts. This technique will be discussed in a second article, a 95 sector GIF file has been successfully implanted in three chained together Extended BASIC programs.

2. OVERVIEW

An Extended BASIC program and a linked assembly program are used to store the D/F 128 file in the program space used by Extended BASIC.

a) In Extended BASIC, the computer is prepared for assembly by executing CALL INIT .

b) Memory space is reserved in an area normally used to store the Extended BASIC program in the 32K RAM by executing: CALL LOAD(-31952,170,0,170,0)

c) A program named MERGE1 is then loaded from disk in MERGE mode. MERGE1 is RUN and it loads an assembly program named GIFO from disk.

MERGE1 prompts for the file name which is to be implanted and loads the file from disk into a variable, one record at a time.

GIFO is used to pack the D/F 128 file records into memory. The space used for the D/F 128 storage is from address >FF97 down as far as is required. Once the file is loaded into the RAM, the pointers to the start and end of the Extended BASIC line number list are adjusted to both point to the byte immediately below the file in memory. The modification of the pointers eliminates MERGE1 which is no longer required.

A small Extended BASIC program (MERGE2) is then merged into memory. MERGE2 controls the re-creation process for the D/F 128 file and provides an Extended BASIC program environment for the implanted file. MERGE2 is edited to give the correct conditions for the process of re-creating the file.

Finally another assembly routine named GIFV is loaded above the file and below the top end of available memory, between >FF98 and >FFE7. GIFV is used to retrieve the D/F 128 file records.

The program (with implanted file) is then saved to disk, ready to be placed on the BBS for the users to download.

3. METHOD

Both instructions and comments are included in this method. The instructions are flagged with a "*" at the commencement of each line.

a) Reserve memory space by adjusting the line table pointers:

```
* CALL INIT :: CALL LOAD(-31952,170,0,170,0)
```

b) Get the first loader:

```
* MERGE DSK1.MERGE1
```

c) Run the just loaded MERGE1.

```
* RUN
```

Respond to the original file name prompt, eg,

```
* DSK1.MOON-GIF
```

The program now loads the D/F 128 file and prints the record number and the current memory addresses for each file record. Values of R, H and L print on the screen during this process.

R is the Record number just implanted.

H is the High byte of the first address of 128 memory bytes just used.

L is the Low byte of the first address of 128 memory bytes just used.

```
* Write down the last R value for use in MERGE2.
```

The final value of H must be greater than 170 and the final value of L must not be zero.

Instructions are then printed on the screen for the next steps in the process and the MERGE1 program wipes itself out by changing the line table pointers.

d) MERGE2 is loaded next:

```
* MERGE DSK1.MERGE2
```

In MERGE2, edit the following lines:

Line 100 - change this line to give a suitable program name when it is saved to disk, eg,

```
* 100 ! SAVE DSK1.G_MOON
```

Line 120 - Edit the resultant file name in the OPEN statement. This is the name of the original D/F 128 file which will be given to the file again when it is un-implanted. It is contained in the quotes after the device name.

```
* 120 OPEN #1:"DSK1.MOON-GIF",OUTPUT,DISPLAY ,FIXED 128
```

Line 130 - Place the last value of R from the running of MERGE1 in the TO value of the FOR-TO-NEXT loop.

```
* 130 FOR R=1 TO 71 :: etc
```

e) Load the assembly routine GIFV:

```
* CALL LOAD("DSK1.GIFV")
```

f) Save the program using line 100, ie 100, FCIN X, ENTER, FCIN 8 (Redo), edit off the "100 !" and press ENTER.

```
* SAVE DSK1.G_MOON
```

The file is now implanted inside the Extended BASIC program. The program should next be run, writing the new file to a different disk from the one from which it originally came. Then load the file into GIF-MANIA and rejoice at your success.

4. PROGRAM NOTES

The un-loader programs for the file represent an overhead which uses additional disk storage space and increases download time from the BBS. The combination of the space taken by the assembly file GIFV and the Extended BASIC program MERGE2 make up this overhead. By using multi-statement lines in MERGE2, it was possible to restrict this overhead to just one sector of disk space.

The program is designed to be loaded and then run ONCE only. If it needs to be run again, load it from disk again, FIRST. The reason for this is that the pointer to the start of the D/F 128 file in memory is only a valid value when it is run for the FIRST TIME AFTER LOADING.

If the file is examined in memory, it will appear jumbled. The first record is written from >FF18 up to >FF98, the second from >FE98 up to >FF18 and so on.

5. PROGRAMS

A total of four programs are used in the process of implanting these files. The CALL INIT :: CALL LOAD (-31952,170,0,170,0) is best executed from the command mode.

The assembly routines are given as source files, they must be assembled before use.

GIFS is assembled as GIFO
GIFU is assembled as GIFV

The Extended BASIC programs should be saved in MERGE mode with Extended BASIC. These programs are named MERGE1 and MERGE2 .

```

-----*-----
Extended BASIC program MERGE1
100 ! SAVE DSK1.MERGE1,MERGE
110 CALL INIT :: CALL CHARPAT(34,A$):: CALL CHAR(128,A$)
120 CALL LOAD("DSK1.GIFO")
130 DISPLAY AT(2,1)ERASE ALL:"Enter DV128 GIF file name
    DSKx.FILENAME"
140 ACCEPT AT(4,1)BEEP SIZE(-16):FNS
150 OPEN #1:FNS,INPUT ,DISPLAY ,FIXED 128
160 IF EOF(1)THEN 200 ELSE R=R+1 :: LINPUT #1:A$
170 CALL LINK("GIFSAV",A$,H,L)
180 PRINT "R=";R;" H=";H;" L=";L
190 GOTO 160
200 CLOSE #1
210 PRINT "The GIF File has been placed"
220 PRINT "in memory, from";H;L;"up."
230 PRINT ":"Merge the loader now using"
240 PRINT "MERGE DSK1.MERGE2"
250 PRINT ":"Edit program name- line 100"
260 PRINT ":"Put the correct end file"
270 PRINT "name in line 120"
280 PRINT ":"Replace TO value in line 130"
290 PRINT "with last R value =";R
300 PRINT ":"Next load the assembly with"
310 PRINT "CALL LOAD(";CHR$(128);"DSK1.GIFV";CHR$(128);
    ")")
320 PRINT ":"Save the resultant implanted"
330 PRINT "file as a program"
340 L=L-1 :: CALL LOAD(-31952,H,L,H,L)

```



* Source=GIFS Object=GIFO Ross Mudie, 1st January 1993

```

DEF GIFSAV
* CALL LINK("GIFSAV",A$,H,L)
STRREF EQU >2014      This program takes a 128 byte file
XMLLNK EQU >2018      record and places it in the appropri-
NUMASG EQU >2008      ate position in memory. It returns
FAC EQU >834A         the lowest address used by each
CIF EQU >20           string as a high and low byte. These
*                   values are ultimately used to modify
SAVRTN BSS 2         the Extended BASIC line number table.
WS BSS 32
D128 DATA 128
STADDR DATA >FF97   Address starts 1 byte low to
*                   allow for length byte for STRREF
GIFSAV MOV R11,@SAVRTN
LWPI WS
S @D128,@STADDR     Gives next address to use
CLR RO              Element zero in variable A$
LI R1,1             Argument 1 in LINK
MOV @STADDR,R2      Destination for string in R2
MOVB @D128+1,*R2    Maximum length byte for STRREF
BLWP @STRREF        Get a string from Extended BASIC
MOVB @STADDR,R3     Get the HIGH byte of the address
SRL R3,8            Make the byte value into a word
MOV R3,@FAC         Put in Floating Point Accumulator
BLWP @XMLLNK        ) Make Integer word into a
DATA CIF            ) Floating Point number for x/b.
CLR RO              Element zero in variable H
LI R1,2             Second argument in the LINK
BLWP @NUMASG        Transfer HIGH byte address value
MOVB @STADDR+1,R3   Get LOW byte of address
SRL R3,8            Make byte into a word
MOV R3,@FAC         Put it in the FAC
BLWP @XMLLNK        Convert Integer of low byte
DATA CIF            address into a full precision number
CLR RO              Element 0 in variable L
LI R1,3             Third argument in LINK
BLWP @NUMASG        Transfer LOW byte address value
CLR RO              )
MOVB RO,@>837C      ) Text book way to return to
LWPI >83E0           ) calling program.
MOV @SAVRTN,R11     )
B *R11              )
END

```

```

-----*-----
Extended BASIC program MERGE2.
100 ! SAVE DSK1.MERGE2,MERGE
110 CALL INIT :: CALL LOAD(8196,255,152)
120 OPEN #1:"DSK1.MOON-GIF",OUTPUT,DISPLAY ,FIXED 128
130 FOR R=1 TO 71 :: CALL LINK("GIFUNS",A$):: PRINT #1:
    A$ :: NEXT R :: CLOSE #1

```

* Source=GIFU Object=GIFV Ross Mudie, 1st January 1993

```

DEF GIFUNS
* CALL LINK("GIFUNS",A$)
AORG >FF98          255,152 Absolute load address
STRASG EQU >2010     When assembled this code loads
XMLLNK EQU >2018     into 80 bytes from >FF98 to >FFE7.
NUMASG EQU >2008
FAC EQU >834A
CIF EQU >20

```

CONTINUED ON PAGE 2.

LEARN TO KNOW YOUR TI
LESSON 2
with Percy Harrison

Well I hope the first lesson was not too boring, they will get more interesting as the lessons progress but it is most essential that you have an understanding of the very basics of the TI language in order to master the more advanced operations of programming.

LESSON 2 COLOUR AND SOUND

This lesson introduces the CALL SCREEN and CALL SOUND statements and makes use of plenty of "bells and whistles" to increase program richness. Let's start.

Enter: NEW

NEW empties the memory and erases the screen. You are now ready to start this lesson.

COLOR THE SCREEN

Enter: 10 REM COLOR THE SCREEN
20 CALL SCREEN(14)

RUN the program.

Line 20 tells the computer to change the screen colour.

RULE: The number in () after SCREEN tells what colour the screen will be. Any number from 1 to 16 is acceptable.

RUN it again trying different numbers.

The colour changes back to the original screen colour as soon as the program is over.

THE COMPUTER PEEPS LIKE A BIRD

Add this line:

30 CALL SOUND(200,1500,15)

RUN it.

Did you hear it "peep"?

If you do not hear a tone, turn up the sound on your TV.

The CALL SOUND command has 3 arguments.

CALL SOUND(length, pitch, loudness)

length 1 to 4250 is the duration of the sound in milliseconds

pitch 110 to 44732 is the pitch in hertz (cycles per second)

loudness 0 (loud) to 30 (off)

Put 500 in place of the 200. Now the peep lasts longer.

Put 800 in place of 1500. Now the sound is lower.

Put 2 in place of 15. Now the peep is louder.

Try other numbers.

The CALL SOUND command will be further explained in a later lesson.

PRINTING AN EMPTY LINE

Run this: 10 REM SOME LINES
20 PRINT "HERE IS A LINE"
30 PRINT
40 PRINT "ONE LINE WAS SKIPPED"

Line 30 just prints a blank line.

STRING CONSTANTS

Look at these print statements:

10 PRINT "JOE"
10 PRINT "#D47ZZ*Z"
10 PRINT "19"
10 PRINT "3.14159265"
10 PRINT "I'M 14"
10 PRINT " "

(Note: Type and RUN each line separately)

Letters, numbers and punctuation marks are called "characters." Even a blank space is a character. Look at this:

10 PRINT " "

Characters in a row make a "string."

The letters are stretched out like beads on a string.

A string between quotation marks is called a "string constant."

It is a string because it is made of letters, numbers and punctuation marks in a row.

It is a constant because it stays the same. It does not change as the program runs.

ASSIGNMENT 2:

1. Write a program that prints your first, middle and last names.
2. Now add a "peep" before it prints each name. Make each "peep" a different tone, deep for your first name, high for your last name.
3. Now make the screen change colour for each name.

Keep practicing, more next month.

ANSWER TO LESSON 1

Assignment Question 1-2

10 REM GREETING
20 PRINT "HI THERE"
30 PRINT "TI COMPUTER"

Nostalgia Time
by Geoff Trott

Last month, I started looking at the earliest copies of our News Digest that I had. Assuming that you found that interesting, I am continuing with that this month. Please stop me if you do not want me to continue. I will repeat the disclaimer from last month in case anyone reads this and gets the wrong idea. I am attempting to describe the look, layout and content of the newsletters without any critical intent. I will try to avoid using any adjectives which could cause offense and if anyone takes offense then that is purely their interpretation of the words and not my intention. I hope that makes it clear and that no one will be offended.

I have looked at the February and April issues of 1983 and shall now move on to the May issue. This is labelled as April also, but as I explained last month, I believe this to be incorrect. It is physically much smaller than the previous issues and looks like it was laid out on A4 sheets and then reduced by 0.7 to give an A5 size. The front cover is white and has the old title at the top as used in the previous 1983 issues but also includes the new logo, which is the one we still use. The magazine had 18 pages with the cover containing just a picture of a satellite between Jupiter and Earth and a headline "Computer Communications in the 80's". Inside there is an article "Profile" from Brian Lewis the Co-ordinator. In it he announces that Specialist Groups are to be set up in May to meet at Merrylands, Macquarie University, North Rocks, Miranda, Collaroy and Bondi. These are clearly the start of the regional groups and some are in private homes while others are in commercial establishments.

John Robinson has a Secretary's News column in which he outlines the activities of the club for the last month and announces the activities in the coming months. Last month, the members voted on the club logo and chose the logo from Manuel Constantinidis which earned for him one year's free membership. There were 61 members and guests in attendance. At the last committee meeting a working Library Committee was formed to look at the software library and index and classify the programs, issue special purpose tapes and ensure a balance between various languages on the monthly tapes. Shane Andersen was voted unanimously a life member of the Club "in recognition as a founder of the Club and the hard and unrewarded work done during the club's early days".

There was also an editorial from Shane with information about a program he had received from Doug Thomas in Melbourne that used a speech synthesizer and Terminal Emulator 2 module to sing a song. There is a short article extolling the virtues of TI-Writer which has no author's name. The centre pages of the issue are by Shane, Andrew Nutting, John Robinson and Alwyn Smith about using modems for communicating around the world. This normally required at this time an RS232 interface, Modem and Terminal Emulator module (plus a bit of money for the phone calls). There is an article by Andrew Nutting showing the connections for cabling a printer to the PIO port of the RS232 card. Younger Set, with Jenny, has a hall of fame for games players and a word puzzle competition.

Programs in the issue include an Extended BASIC program to help with taxation by Neil Sakac, Jumping Jack in BASIC by Brett Pijan and an assembler program to clear the screen in a curtain effect by Manuel Constantinidis. There is a hint on how to get TE#2 and the speech synthesizer to speak the listing of your program and how to convert a 115 volt peripheral expansion box to 240 volt operation by Russell Welham.

The June 1983 issue, has the size and format that we know so well. It has pictures and an index on the cover which is white, is 12 pages long and has general club information along with mailing information on the back cover. Inside, in the editorial, Shane explains that they have been experimenting with various sizes to find the best one for printing and posting and have decided that this is the best. Secretary John Robinson re-enforces that and notes that there are 340 members receiving the News Digest. (It did not return to that small a number until 1987.) John also notes that the high speed tape recorder has been stolen and police have been investigating. He was receiving about 40 letters a week and mentions one who was interested in "applying the computer to various facets of a model railway". Derek Harns should now contact Ross Mudie! The June meeting is to be an all day tutorial/workshop. Members bring along their systems, an extension cord, a card table and some food to put into a pot luck lunch. There was a good range of activities organised with people like Russell Welham taking a class on programming music, for example, and a games play-off for the younger set. It is noted that Antony Lewis, who was the club librarian, had to relinquish that post while he concentrated on his studies and Manuel Constantinidis had taken over.

There is an historical article from TI's Home Computer Service Club Newsletter, detailing the inventions due to engineers at TI. Peter Lynden has an article on Logo 2 in an educational sense as an excellent language for children to learn as they become familiar with computers. Peter also wrote a short article on making TI-Writer easier to use. There is also an article by David Liell on how to debug programs. He suggests that you should always start a program with a REM statement with information about who wrote the program, the date and any special requirements. Then, anywhere you do something tricky (as in hard for someone else to understand) include a REM explaining what you are doing. David then goes on to talk about finding a bug in a READ statement and its associated DATA statements. These can be a problem as you can miss a comma or insert a comma and then have the wrong number of items in the DATA statement for the READ statement.

The programs in the issue include an assembler language program from Geoff Patterson to enable a small character set to be installed and a BASIC program to call it. There is a BASIC program to catalog a disk, a Motorcross game by Jane McAshan in BASIC, a program to print a BASIC program from a "list" file by Geoff Patterson, Moon Lander game converted to Extended BASIC by Russell Welham and a short note showing how to list a BASIC program to expansion memory using the MiniMemory module and then to display the listing using a Memory Dump program.

Well, that covers the next two magazines and I really enjoyed reading them for you. If you want more information about any of the articles or programs mentioned, contact me somehow and I will do my best. You could also read the newsletters in the club's library.

MAKING A RECORDING VOLTMETER
USING METEX MULTIMETER
AND
A TI99/4A COMPUTER

by Ross Mudie, 26th January 1993.

In a previous article I described an assembly program and the wiring needed to connect the 1200 baud serial data output of a METEX M-3650CR multimeter into a TI99/4A computer. This article discusses using a time of day clock in the computer and the METEX interface program to make a recording voltmeter.

The program is written in Extended BASIC, using the assembly language program described in the previous article. It is intended that this should promote ideas so that others can write their own Extended BASIC program to meet special needs. The program presented loads the assembly language program and then checks that the multimeter is both connected and turned on. The option is given to record the readings in a disk file or just on the computer screen. Readings are taken from the multimeter as rapidly as possible. Between 60 and 64 readings per minute are obtained. This is limited by how much processing time is taken in the Extended BASIC program. The program records the minimum, maximum, average and the number of readings taken each minute.

The file names use a combination of the date and time from the clock to prevent a new file from overwriting a previous file. The file name DSK1.26/01:1932 would mean that the program commenced running at 7.32pm on 26th January. Long variable names are used which describe the function of a number of the variables. This reduces the need for remarks and gives a program which is self-documenting. The program will be available for download on TEXPAC BBS until the end of March 1993.

The clock used is in a Techo Time article by John Paine on page 5 of the March 1988 TND.

The following shows a sample of the disk log file produced by the program whilst monitoring the nominally 240 volts mains supply with the Metex multimeter.

Please Note: If you do not know or understand the safety precautions necessary when working with potentially lethal mains power voltages, then for safety sake, do not measure or work with the power mains.

The column headings in the METEX DISK LOG have the following meanings:

- TIME- The Hour and Minute of the reading which was taken over the previous sixty seconds.
- QTY - The number of readings taken in the previous sixty seconds.
- AVG - The average of the readings of the last sixty seconds.
- MIN - The lowest voltage recorded in the last sixty seconds.
- MAX - The highest voltage recorded in the last sixty seconds.

METEX DISK LOG

File name DSK1.26/01:1802

Date=26/01/1993 Start time= 1802:38

TIME	QTY	AVG	MIN	MAX
1804	61	237.7	235	240
1805	62	237.7	235	239
1806	62	236.7	235	238
1807	61	237	235	239
1808	61	234.9	231	239
1809	61	231.9	230	234
1810	61	234.6	233	236
1811	62	236.3	235	238
1812	63	236.2	234	237
1813	63	237.3	236	238
1814	63	235.9	234	237
1815	63	234.2	233	236
1816	62	238.9	234	240
1817	63	240.6	239	242
1818	63	240.7	239	243
1819	63	240.3	238	244
1820	62	239.6	238	242
1821	63	242.2	241	245
1822	63	243.7	243	245
1823	63	242.8	242	245
1824	63	243.7	242	245
1825	63	244	243	245

Extended BASIC program:

```

100 ! SAVE DSK1.1MINMETEX1
110 DD$="DSK1." :: YEAR$="/1993" ! Specify disk drive and Year
120 DEF DECIMAL=D-6*INT(D/16)! BCD to decimal converter
130 CALL CLEAR
140 CALL SCREEN(6):: FOR S=0 TO 12 :: CALL COLOR(S,16,1)
    :: NEXT S
150 DISPLAY AT(1,1):"Loading Metex assembly": "IS METER
    CONNECTED and ON?"
160 CALL INIT
170 CALL LOAD("DSK1.METEX")
180 OPEN #1:"RS232.TW,DA=7.PA=N.BA=1200"
190 DISPLAY AT(1,1):"Assembly is loaded": "IS METER CON
    NECTED and ON?"
200 CALL LINK("MXREAD",METEX$)
210 DISPLAY AT(1,1)BEEP: "METER IS CONNECTED and ON":
    "LOG RESULTS TO DISK? Y/N?"
220 CALL KEY(3,K,S):: IF K=ASC("Y")THEN DISKLOGON=1 ELSE
    IF K>ASC("N")THEN 220
230 IF DISKLOGON THEN DISPLAY AT(5,23):" " ELSE DISPLAY
    AT(5,22)SIZE(2):" "
240 GOSUB 510 :: GOSUB 580
250 IF DISKLOGON THEN DISKLOGNAME$=DD$&SEG$(DATE$,1,5)&
    "&SEG$(TIME$,1,4)
260 IF DISKLOGON THEN OPEN #2:DISKLOGNAME$
270 DISPLAY ERASE ALL:"Reading Metex": "Hold SPACE to e
    nd program after first reading which will be aft
    er the first full minute."
280 IF DISKLOGON THEN DISPLAY "Disk file ";DISKLOGNAME$
    "will be created" ELSE DISPLAY "NO diskfile will b
    e created"
290 DISPLAY "TIME- Hour/Minute printed": "QTY - Number
    of readings": "AVG - Average of readings": "MIN -
    Lowest reading taken": "MAX - Highest reading take
    n"
300 DISPLAY "Date=";DATE$;" ";TIME$&";";SECOND$: :
310 IF DISKLOGON THEN PRINT #2:"METEX DISK LOG": "File
    name ";DISKLOGNAME$: "Date=";DATE$;" Start time="
    ;TIME$&";";SECOND$: :
320 DISPLAY "TIME QTY AVG MIN MAX"
330 IF DISKLOGON THEN PRINT #2:"TIME QTY AVG MIN MAX
    "
340 GOSUB 580 :: OLDMINUTE=MINUTE
350 GOSUB 580 :: IF OLDMINUTE=MINUTE THEN 350 ! Wait unt
    il new minute
360 OLDMINUTE=MINUTE
370 CALL LINK("MXREAD",METEX$)
380 CALL KEY(5,K,S):: IF K=32 THEN 680
390 READING=VAL(SEG$(METEX$,5,3))
400 R=R+1 :: RDT=RDT+READING

```

CONTINUED ON PAGE 22

WORD PROCESSING
PART 3
by Col Christensen
Brisbane User Group

Part 3 covers file saving files to disk, loading them from disk and printing text files from the memory buffer. All these are performed from the Command made with simple two-letter commands. We also look at some of the Formatter dot commands.

FILE HANDLING

PRINTING FILES

PF is the command to send the text in the memory buffer to a printer. You are prompted for a devicename for your printer. Valid devicenames normally used are PIO (Parallel Input/Output) if you have a parallel printer or RS232 if using the less common type of printer with a serial input. In this discussion I shall use the devicename, PIO, as hardly anyone today is using a serial printer.

So for printfile devicename, making the entry, PIO, followed by <ENTER> will print out the whole of the text in memory just as it would appear on the screen except for printer codes. The topic of printer codes will be covered in a later issue.

But the WP is more versatile than just being able to print out the complete text buffer. You can direct the printout to cover a specific part of the whole file by preceding the devicename with numbers or letters. The following are valid input for the devicename, PIO.

1 50 PIO prints only lines from 1 to 50.

48 E PIO prints lines 48 through to the end of the text. Remember the "E" represents the last line number.

L PIO prints the line numbers as well as the text but only the first 74 characters of text on each line. A line would normally be up to 80 characters long but the other 6 are taken up by the 4 digits of the line number and the two spaces following it.

L 33 46 PIO prints (can you guess?), yes, the lines from 33 to 46 including the line numbers. Amazing!

F PIO prints the text in fixed 80 format. One use for this function is to edit an assembly object code file and to save it back to disk by using a disk filename instead of a printer devicename. WP text files are normally saved as VARIABLE length records.

C PIO strips out any control characters like carriage returns, line feeds or new page before printing. I have never used this so I must try it out. Maybe you can think of a good use for it.

DELETING FILES

P purges all text in the memory buffers if you answer "Yes" to the prompt. You then have a clean slate again to start on.

DF for delete file appeared in the TI-Writer Editor so that particular disk files could be deleted. Its function has been greatly superceded and enhanced in Funnelweb's word processor to become a SD (show directory) command. You need to consult the Funnelweb docs to discover all its finer points of disk and file management.

SAVING AND LOADING FILES FROM DISK

Yes, SF is the abbreviated command name for this utility. A prompt then requires the devicename to save the file to. There I go again. I am sure a preposition is not the correct thing to end a sentence with. There are additional refinements in this command like those above for printing files. You can save parts of your text by preceding the devicename with line numbers. Here are some valid entries:

DSK1.FILENAME
1 108 DSK1.FILENAME
26 E DSK1.FILENAME

To load a file, type LF on the command line and then at the prompt the pathname to the file on disk such as:

DSK1.FILENAME
16 120 DSK1.FILENAME loads in just those line numbers from the disk and stores them in the memory buffer numbered from line 1, of course.

O 40 DSK1.FILENAME
100 E DSK1.FILENAME

Things now get deeper. You can merge a disk file with what is already in memory with:

10 DSK1.FILENAME merges all of DSK1.FILENAME and places it after the existing line 10 and before the existing line 11. Obviously the old lines 11 onwards will now have much higher line numbers.

E DSK1.FILENAME merges all the disk file at the end of the existing file in memory.

O DSK1.FILENAME merges all the disk file at the beginning of the existing file in memory.

And deeper yet! Merge PART of a disk file into the existing text with:

14 50 64 DSK1.FILENAME. It should be no trouble to work that one out if you remember that the first number is the line in the current memory buffer after which the insertion is to be made.

SEARCH

This group allows finding a particular "word" or replacing it with another.

FS (FindString) is followed by a prompt to input the string or word for which a search is to be made. The string must be preceded by a slash and followed by a slash. For example, to find the string "word processor" your entry would be /word processor/. The search is case sensitive and will not find a string in upper case (WORD PROCESSOR) when the input is in lower case (/word processor/). After pressing <ENTER>, the text will be searched from the position of the cursor onwards. If you want the whole lot to be searched, do a Show Line 1 first to get the cursor to the beginning. The search will end if successful with the cursor over the first character in the first matching string found. If not successful, the cursor will appear after the end of the last line of text.

RS (ReplaceString) requires an input of the string to be searched for as well as the string to replace it with. An input such as /RSI/Repetitive Strain Injury/ will replace the abbreviation with the full name for the "affliction". But when the cursor appears over the first occurrence of the string "RSI" this list of prompts can be seen on the command line -

Yes, No, All or Stop?

So type
Y to replace this one and find the next.
N to ignore this one and go on to the next.
A to replace all occurrences of the string.
S to stop and escape to the edit mode.

NOTE WELL. If in the word-wrap mode with the solid cursor, all string replacements will be accompanied by automatic reformatting of that paragraph using the current tab settings. This can be disconcerting if a particular replacement is in a section of your work that has been set out in tabular form. Reformatting will close it all up to just one space between each item. To overcome this, turn word wrap off before making the change in that section.

TEXT FORMATTER

The text formatter is a program loaded separately which allows the printing of a text file according to the formatting commands that are imbedded in the text file. These commands in the text file referred to as dot commands are the ones that set the current left and right margins, paragraph indentation, page length and line spacing etc. for printing.

When the formatter program is loaded, the first prompt asks for the INPUT FILENAME. This is the name under which the text file had been saved and will be the name of the file you want to print.

The second prompt of PRINT DEVICENAME will usually be answered as PIO.LF unless you have a serial printer. Then RS232.LF will be the devicename but consult your printer manual in case some special baud rate is needed in the serial devicename. In either case the formatter itself will issue line feed commands to the printer at the required places. To all the other prompts on the formatter screen, just press <ENTER> for now to accept the defaults shown and printing should begin.

THE FORMATTER COMMANDS

The formatter commands, are always in upper case and preceded by a dot (hence the name dot commands) and placed at the appropriate positions in the text on a line by themselves. More than one command can be placed on a line. There needs to be only one dot used only at the beginning of a line and each command must be separated by a semicolon. e.g. The very first line in a text file might have the following formatter commands:

.LM5;RM75;IN+5;FI;AD;PL56;CE2

It is also permissible to leave spaces between the command and the number parameter following it, such as-

.LM 5;RM 75 etc. The following explanations will throw some light on how these commands will influence the final printed output.

FORMATTER TEXT DIMENSION COMMANDS

LEFT MARGIN

LM 6 sets the left margin at column 6 on the page.

LM +5 adjusts the left margin inwards 5 columns more than the previous setting.

LM -5 adjusts the left margin outwards 5 columns less than the previous setting.

RIGHT MARGIN

RM 70 sets the right margin at column 70 on the printed page.

RM +5 adjusts the right margin outwards 5 columns more than the previous setting.

RM -5 adjusts the right margin inwards 5 columns less than the previous setting.

INDENT

IN 8 indents the first line after a Cr (carriage return) to column 8.

IN +10 sets the indentation to 10 columns inwards from the current LM setting.

IN -5 sets indenting to 5 columns less than the LM setting.

LINE MANIPULATION

FI (Fill) puts as many whole words as possible on each line to fill within the left and right margin limits.

NF (No fill) cancels the FI command and prints the part of the document following the NF exactly as it would appear on the screen.

AD (Adjust) in conjunction with a Fill command spreads the spacing between words so that the printed text reaches the right margin exactly. Lines ending with a Cr will not be adjusted nor will they need to be. The formatter cannot differentiate between printer control codes placed within the text and ordinary text, so adjusted lines containing control codes may not completely reach the right margin.

NA (No adjust) cancels the adjust command and the printout following it will have raggedy ends.

LINE SPACING

LS 2 causes printing on every second line only.

PAGE LENGTH

PL 60 prints 60 lines then starts a new page.

PL +5 adjusts the page length relative to the previous setting

PL -4 as above

BEGIN PAGE

BP forces a new page break. The current PL value is then restored and countdown starts again.

INTERNAL FORMAT COMMANDS

CENTRE TEXT

CE centres the next line between the current left and right margins.

CE 3 centres the next 3 lines.

SPACE

SP leaves one blank line on the printed page.

SP 5 leaves 5 blank lines.

The above will cover the basic intricacies in using formatting commands to manipulate the way in which some text is printed. Remember two things:

1. The Tab settings govern the way the text will look on the screen and the dot commands in the text will control the way the text is printed.

2. On a line of formatter dot commands, an INdent with + or - relative positioning, relates to the last mentioned LM setting. On a line like the example earlier where many dot commands are placed, it is good practice to place an LM first if making changes to this margin.

I guess by now you have discovered what the given one-line formatting example above will do.

In Part 4, I shall introduce Formatter commands and other methods to induce underlining, overstriking, transliterates, page identification and more file management.

TiSHUG Software
by Larry Saunders

Software releases for December

This is the first part of a complete series of the best of the TI games. The loader is a RAM style loader that is very user friendly. To change colour, PRESS V, to save colour choice PRESS FCTN 5, FCTN 9, Space Bar.

Note: Ms Pacman has a fault. DO NOT eat the FRUIT after about the third lot or else it will LOCK-UP. Connect Four is the same as the board game except it has a extra game called DROPOUT. Dropout gives you the choice of dropping your chip in the top or removing one of your chips from the bottom line only. If select bottom all chips in that line move down one space.

G001

Used= 355 Free= 3

CHARA1 9 Prog
CONNECT4 13 Prog
CONNECT5 26 Prog
LOAD 5 Prog
MIDN 33*Prog
MIDO 2*Prog
MINE 33*Prog
MINF 33*Prog
MING 2*Prog
MOON 33*Prog
MOOO 33*Prog
MOOP 16*Prog
MS-P 33*Prog
MS-Q 30*Prog
ROOT 28 Prog
ROTO/RAID 26*Prog

G002

Arcturs two joysticks required.

Used= 347 Free= 11

ARCTURS1XB 36*Prog
ARCTURS2XB 41*Prog
ARCTURS3XB 31*Prog
BARRAGE1XB 35*Prog
BARRAGE2XB 18*Prog
BUCK2RG1XB 35*Prog
BUCK2RG2XB 43*Prog
BURGER1XB 33*Prog
BURGER2XB 33*Prog
CHARA1 9*Prog
LOAD 5*Prog
ROOT 28 Prog

AR005 is a collection of TI Artist Fonts never before release in TI Artist format. Converted by ALF.

Used= 358 Free= 0

ATHEN_F 30 d 80
BANNER_F 59 d 80
BSBOLD_F 31 d 80
CASLOW_F 34 d 80
CURSIVE_F 74 d 80
FANCY_F 42 d 80
FAST_F 26 d 80
FILM_F 62 d 80

G006

Pinball requires two joysticks.

Used= 357 Free= 1

CHARA1 9 Prog
JUMPYXB 39 Prog
LOAD 5 Prog
PARSEC1 33*Prog
PARSEC2 33*Prog
PARSEC3 26*Prog
PARSEC4 19*Prog
PINB 33*Prog
PINC 13*Prog
RIVER 33*Prog
RIVES 18*Prog
ROOT 28 Prog
SUBCOM 33*Prog
SUBCON 33*Prog
SUBCOO 2*Prog

Prog U003 is a very usefull disk of utility programs. Ros configure V7.35, DM 1000 V5, Editor Assembler that looks like the module and runs most programs that need the module. (It will not run ROCK RUNNER). Keyboard/Joystick tester, Archiver, Nibbler.

U003

Used= 301 Free= 57

ARC3 33 Prog
CFG 24 Prog
CHARA1 9 Prog
DM 33 Prog
DN 31 Prog
ED/AS 33 Prog
KJTESTXB 14 Prog
LOAD 5 Prog
MENU 31 Prog
NIBBLER 27*Prog
ROOT 28 Prog
ROS 33 Prog

P004 is a large collection of Page Pro Small Pictures that have never been release in Page Pro format before. These were converted from my collection by Alf Ruggeri

Used= 323 Free= 35

L/AIRPLAN	2 I 13	L/HMAKER	3 I 13
L/APPLE	2 I 13	L/HORSES	3 I 13
L/ARK	3 I 13	L/HOUSE	2 I 13
L/ASTER	3 I 13	L/HOUSE2	3 I 13
L/AUTOS	3 I 13	L/HOUSE3	3 I 13
L/BACKHOE	3 I 13	L/HUNTING	3 I 13
L/BANK	3 I 13	L/LLUVYOU	3 I 13
L/BASBALL	3 I 13	L/JACK	3 I 13
L/BFLY	2 I 13	L/LARKSPR	3 I 13
L/BIKE	2 I 13	L/LAWYER	3 I 13
L/BLIMP	2 I 13	L/LECTRIC	3 I 13
L/BOAT	2 I 13	L/LEXIT	2 I 13
L/BOWLER	3 I 13	L/LINER	3 I 13
L/BSKBALL	3 I 13	L/MECANIC	3 I 13
L/BUNNY1	2 I 13	L/MONITOR	3 I 13
L/CABOOSE	2 I 13	L/MOOSE	3 I 13
L/CAKE	2 I 13	L/MR-DISK	3 I 13
L/CALENDU	3 I 13	L/MUG	3 I 13
L/CAMEL	3 I 13	L/MUSIC	3 I 13
L/CANDLE	2 I 13	L/NOSIGN	3 I 13
L/CANDLE2	3 I 13	L/NOTES	3 I 13
L/CAR	2 I 13	L/NURSE	3 I 13
L/CARNATN	3 I 13	L/OAKTREE	3 I 13
L/CAT	3 I 13	L/OHIO	3 I 13
L/CHRYSAN	3 I 13	L/PHONE	3 I 13
L/CHURCH	3 I 13	L/PHOTOG	3 I 13
L/CIGAR	2 I 13	L/PIANO	2 I 13
L/CIN-DAY	3 I 13	L/PIANO2	3 I 13
L/COMPUTR	3 I 13	L/PLUMBER	3 I 13
L/CPLOGO	3 I 13	L/POINSET	3 I 13
L/DAFODIL	3 I 13	L/POPPY	3 I 13
L/DENTIST	3 I 13	L/PUMPKIN	2 I 13
L/DISK	2 I 13	L/REALTOR	3 I 13
L/DOCTOR	3 I 13	L/REXIT	2 I 13
L/DOZER	2 I 13	L/ROSE2	3 I 13
L/DRAFTIN	3 I 13	L/ROSE3	3 I 13
L/DUTROW	3 I 13	L/RRCROSS	2 I 13
L/EBUNNY	2 I 13	L/SCISSOR	2 I 13
L/ENGINE2	2 I 13	L/SECTARY	3 I 13
L/FISHING	3 I 13	L/SFLAKE1	2 I 13
L/FIXIT	3 I 13	L/SFLAKE2	2 I 13
L/FLORIDA	3 I 13	L/SKIBUM	3 I 13
L/FLOWER1	3 I 13	L/SKULL	3 I 13
L/FLOWER2	2 I 13	L/SMILE	3 I 13
L/FLOWER3	3 I 13	L/SNOWMAN	3 I 13
L/FLOWER4	3 I 13	L/STOPLIT	2 I 13
L/FOOTBAL	3 I 13	L/SWAN	3 I 13
L/FORCES	3 I 13	L/TEACHER	3 I 13
L/GARDENR	3 I 13	L/TI-99	3 I 13
L/GASPUMP	3 I 13	L/TREE	2 I 13
L/GOLFER	3 I 13	L/VASE	2 I 13
L/GOLIT	2 I 13	L/WREATH	2 I 13
L/GYM1	3 I 13	L/XMSTREE	2 I 13
L/GYM2	2 I 13		
L/GYM3	2 I 13		
L/GYM4	2 I 13		
L/GYM5	2 I 13		
L/HBLAB	3 I 13		
L/HEART	2 I 13		

G007

Bigfoot, once all items are collected around the screen, to move to next level, climb up on the right side off Bigfoot and lower the cage over him, then raise the cage.

Used= 333 Free= 25

BIGFLOAD	2*Prog
BIGFOOT1XB	44*Prog
BIGFOOT2XB	43*Prog
CHARA1	9*Prog
DONKEY1XB	35*Prog
DONKEY2XB	18*Prog
LOAD	5*Prog
METEORB1XB	36*Prog
METEORB2XB	35*Prog
METEORB3XB	25*Prog
PACMAN1XB	35*Prog
PACMAN2XB	18*Prog
ROOT	28 Prog

TiSHUG Software
by Larry Saunders

Software releases for February, 1993

This is the second part of a complete series of the best of the TI games. The loader is a RAM style loader that is very user friendly. To change colour, PRESS V. To save colour choice PRESS FCTN 5, FCTN 9, Space Bar.

G008

MISSION is a war plane fighting game that you take off from a airport, fly over ocean and land, avoid the bombs, etc. and destroy as many targets as possible. Watch out for attacking aircraft.

POKER SOLITAIRE. A card game that will get you in.

POLE POSITION. Max speed is 195 MPH. Joysticks are recommended to steer the car, and keyboard to increase/decrease speed, and shift gear. Increase speed E or I. Decrease speed X or M. Gear shift Q or Y or V. I like using Q, E, & X keys. Other keys are Left S or J. Right D or K. Upper-Left W or U. Upper-Right R or O. Lower-Left Z or N. Lower-Right C or a ', '. Always change to high gear at 100 MPH.

TI-RUNNER is a very good game that has a demo game for the first screen. You can choose the level that you want to start. FCTN 5 is used to page through the levels.

SNAKES AND LADDERS is now pre-scanned as well as every other Extended Basic program that I am releasing as Club Software disks.

Used= 349 Free= 9

CHARA1	9*Prog
LOAD	5*Prog
MISSION	33*Prog
MISSION	32*Prog
POKER/SOL	25*Prog
POLEPOS1XB	35*Prog
POLEPOS2XB	34*Prog
ROOT	28 Prog
RUNNER1XB	28*Prog
RUNNER2XB	35*Prog
RUNNER3XB	43*Prog
SNAKES&LAD	40*Prog
SNAKESLOAD	2*Prog

G009

BARRAGE is a game that you try to protect the ground from falling objects. Joysticks are recommended.

BOXES: Match up the hidden items. One or Two players can play.

HOPPER: Avoid getting caught. Punch bails around the screen. Kill clowns with bails.

JUNGLE HUNT: Jump from vine to vine, swim the river. Watch out for the Crocks. Run up the hill. Watch out for the rocks. Rescue the maiden from the pot. Watch out for the natives. They want to keep her.

MOSAIC PUZZLE. Put the squares back in the right order.

SPACE STATION PHATA: A space maze program.

XORKLE (two versions) 5000 Points or 10000 Points. Rules are supplied in game program. Up to Six players can play.

Used= 354 Free= 4

BARRAGE	33*Prog
BARRAGE	18*Prog
BOXES	24 Prog
CHARA1	9*Prog
HOPP	23*Prog
JUNGLEH1XB	35*Prog
JUNGLEH2XB	34*Prog
LOAD	5*Prog
MOSAICPUZL	24*Prog
PHETA1XB	35*Prog
PHETA2XB	22*Prog
ROOT	28 Prog
XORKLE	32*Prog
XORKLE5	32*Prog

AV010

Adventure Games disk base module, with a selection of Adventure Games.

Used= 323 Free= 35

ADVENTUR_1	13 Prog
ADVENTUR_2	25 Prog
CAVE1	53*Prog
CAVE2	43*Prog
CAVEINFO	11d 80
GHOSTTOWN	41*Prog
IRONHEART	50*Prog
LOAD	6 Prog
SORCERER	39 Prog
SPIDEY	42*Prog

PPO11

The first set of archive PICS pictures in PAGE PRO format. GRAB 1 to 126. Root, Archiver, and PicDimision is supplied with this program.

Used= 350 Free= 8

ARC	33 Prog
GRAB01-42	73 I128
GRAB43-84	78 I128
GRAB85-126	78 I128
LOAD	5*Prog
PICTDIM	23 Prog
PICTDIMDOC	32 d 80
ROOT	28 Prog

Software releases for March, 1993

This is the third part of a complete series of the best of the TI games. The loader is a RAM style loader that is very user friendly. To change colour, PRESS V, to save colour choice PRESS FCTN 5, FCTN 9, Space Bar.

G012

AMAZEING- a maze game, one person game.

BLASTO TANK GAME, try to clear the screen before time runs out, one or two players.

CARWARS, very fast, one player against the computer, try to get all the dots on the grid with out crashing into the computer car.

CROSSFIRE, a space type shootup game.

FACE MAKER, a face making program and game.

FATHOM, a game that is hard to play, but if can is very good.

Used= 353 Free=5

AMAZEING1	13*Prog
AMAZEING2	25*Prog
BLASTO_1	13*Prog
BLASTO_2	25*Prog
CARWARS1	33*Prog
CARWARS2	11*Prog
CROSSFIRE	23*Prog
FACE	33*Prog
FACF	33*Prog
FACG	10*Prog
FATH	33*Prog
FATI	33*Prog
FATJ	33*Prog
FATK	2*Prog
LOAD	5 Prog
ROOT	28 Prog

G013

DRIVE DEMON, a car racing game that is very good but very hard to play.

HANGMAN, a classic game.

HUSTLER, a horse racing game.

NUT-Z, a game that you have to jump around the screen from one block to another, watchout for the arrows.

SUB MARINE COMMAND, a sub program, sink the most tonage without sinking your self. Tennis, another classic game.

TI SQUARES, (basic), one or two player, a classic game that most kids would have played on paper, join the dots and try to get the most squares.

Used= 341 Free=17

DRIVEDEMON	32*Prog
HANGMAN1	13*Prog
HANGMAN2	25*Prog
HUSTLE1	13*Prog
HUSTLE2	25*Prog
LOAD	5 Prog
NUT-Z	30*Prog
ROOT	28 Prog
SUBM	33*Prog
SUBN	9*Prog
TENNIS	33*Prog
TENNIT	33*Prog
TENNIU	33*Prog
TI-SQUARES	27*Prog
TI/1	2*Prog

AV014

The Tunnels of Doom disk base module program, plus some doom games. Editor Assembler required.

AV014

Used= 311 Free=47

ASSAULT	52 Prog	ATC	52 Prog
ATC/INS	8 Prog	LOAD	11 Prog
READ/ME	4 d 80	TOD1	33 Prog
TOD2	33 Prog	TOD3	33 Prog
TOD4	33 Prog	TUNNELS	52 Prog

Used= 324 Free=34

ASSAULT	52 Prog
ATC	52 Prog
ATC/INS	8 Prog
LOAD	11 Prog
READ/ME	4 d 80
TOD1	33 Prog
TOD2	33 Prog
TOD3	33 Prog
TOD4	33 Prog
TOMB	52 Prog
TOMB/INS	13 Prog

P015

The second of the series of TIPS pictures in Page Pro format.

Used= 357 Free=1

ARC	33 Prog
GRBC01-42	81 I128
GRBC43-84	79 I128
GRBC85-126	76 I128
LOAD	5 Prog
PICTDIM	23 Prog
PICTDIMDOC	32 d 80
ROOT	28 Prog

TREASURER'S REPORT

by Cyril Bohlsen

Welcome to my first report as Treasurer,
This will be a very short report, as all my reports will be.

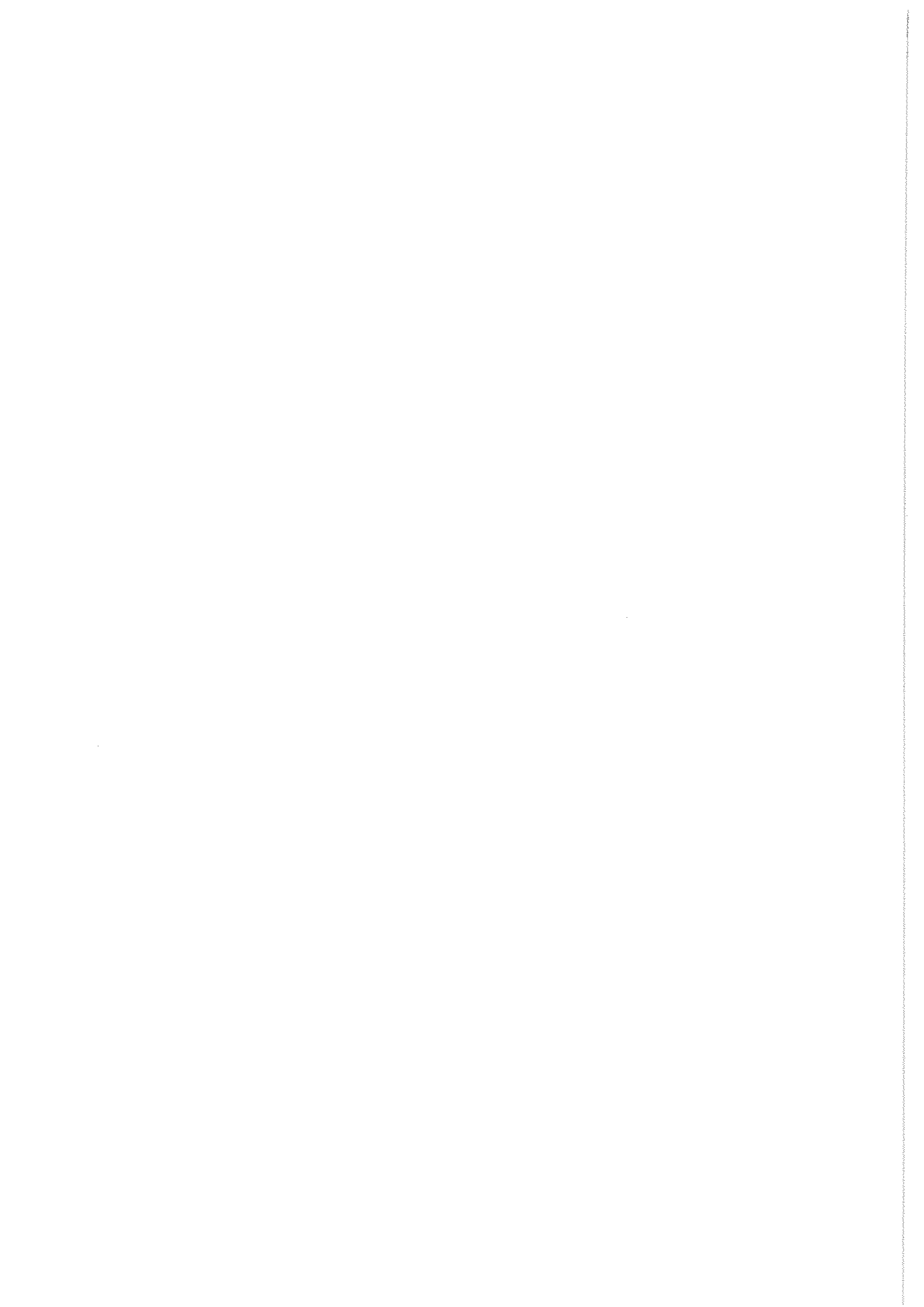
Income for February _____ \$ 1404.70
Expenditure for February _____ \$ 1383.65
this includes some payments for December

Profit for the month _____ \$ 21.95

On behalf our Secretary,
Let me welcome our three new members :-

Leslie Doyle of Tamworth
William Janssen of Toronto
Jim Stylianou of Smithfield

May you have a long and enjoyable association with the club.



When the FILESPLIT program is run, the values of F and R are displayed on the screen. F shows which OUTPUT FILE is being processed whilst R shows which RECORD. Note the value of R achieved in each case. In the case which was used for the development of the system, the file named DREAM-GIF, the values of R were 66, 66 and 55. These R values will be on the screen each time MERGE1 is run during the implant process. If slitting a file into two parts, change the TO value in line 150 from 3 to 2. This will avoid an error message.

4. PREPARING THE LOADERS

In the previous article the program MERGE2 was used to provide the Extended BASIC header and environment for the implanting. MERGE3 was developed from MERGE2, the only difference being the inclusion of line 140 which chains on to the next program.

```
100 ! SAVE DSK1.MERGE3,MERGE
110 CALL INIT :: CALL LOAD(8196,255,152)
120 OPEN #1:"DSK1.DREAM-GIF",OUTPUT,DISPLAY ,FIXED 128
130 FOR R=1 TO 66 :: CALL LINK("GIFUNS",A$):: PRINT #1:
    A$ :: NEXT R :: CLOSE #1
140 PRINT "Running DSK1.G_DREAM2XB" :: RUN "DSK1.G_DREA
M2XB"
```

The next part becomes a little more tricky. Initially I tried to open the file in APPEND mode, however APPEND mode is only applicable with VARIABLE records and GIF is stored in FIXED mode. (Page 139 of the Extended BASIC manual.) To get around this problem, the file in MERGE4 and MERGE5 is opened with RELATIVE organisation. REC is used to specify the RECORD number in line 130 for MERGE4 and MERGE5 to write new records after those written by MERGE3 and MERGE4 respectively.

Here are the programs MERGE4 and MERGE5:

```
100 ! SAVE DSK2.MERGE4,MERGE
110 CALL INIT :: CALL LOAD(8196,255,152)
120 OPEN #1:"DSK1.DREAM-GIF",DISPLAY ,RELATIVE,FIXED 128
130 FOR R=66 TO 131 :: CALL LINK("GIFUNS",A$):: PRINT #1
,REC R:A$ :: NEXT R :: CLOSE #1
140 PRINT "Running DSK1.G_DREAM3XB" :: RUN "DSK1.G_DREAM
3XB"
```

```
100 ! SAVE DSK2.MERGE5,MERGE
110 CALL INIT :: CALL LOAD(8196,255,152)
120 OPEN #1:"DSK1.DREAM-GIF",DISPLAY ,RELATIVE,FIXED 128
130 FOR R=132 TO 187 :: CALL LINK("GIFUNS",A$):: PRINT #
1,REC R:A$ :: NEXT R :: CLOSE #1
140 PRINT "*** Task complete ***"
```

Note the value ranges for R in line 130 in both cases. When working with RECORD numbers in RELATIVE files, the first RECORD number is ZERO. This reduces the FOR and TO values of R by ONE in each case and very nicely gets around the problem of not being able to use APPEND.

5. IMPLANT METHOD

The method used is just a variation of that explained in the article for a one part file. Refer to my earlier article for the detailed commentary.

PART 1 | PART 2 | PART 3

In each case:

```
----- CALL INIT :: CALL LOAD(-31952,170,0,170,0) -----
----- MERGE DSK1.MERGE1 -----
----- RUN -----
```

File names:

```
DSK1.1 | DSK1.2 | DSK1.3
MERGE DSK1.MERGE3 | MERGE DSK1.MERGE4 |MERGE DSK1.MERGE5
Modify merged programs in line 100 after: '100 ! SAVE'
DSK1.G_DREAM1XB | DSK1.G_DREAM2XB | DSK1.G_DREAM3XB
```

----- Line 120, Check and CORRECT ULTIMATE FILE NAME -----

Line 130:

Correct TO value | Correct R=? TO ? | Correct R=? to ?

Line 140 in two places each:

Correct .xxx2XB | Correct .xxx3XB | No change.

In each case load GIFV:

```
-----CALL LOAD("DSK1.GIFV")-----
```

Save each program as it is finished using line 100.

6. TESTING

After all 3 programs have had their respective implants run the first one, note correct chaining to the second and third programs. On successful completion, load GIF-MANIA then load the image file and rejoice at your success.

TIshUG SHOP.
with Percy Harrison.

As promised, this month I am listing the modules that are available from the shop. Over Christmas there has been an unusual demand for modules, mainly from the newer members, and this has left us rather depleted on stocks, however we are continually purchasing modules and so, hopefully, we will have a better range in the near future.

At the moment I am trying to get hold of the instruction books for the following modules and so I would be very grateful if any member who has these books would be prepared to loan them to me for a week or two. They will be returned in the same condition as received.

The booklets that I would like to borrow are:

Hustle
Mind Challengers
Number Magic
Tax/Investment Record Keeping.

TI MODULES AVAILABLE

A-MAZE-ING Module + Book	\$8
CART-WRITER Module Only	\$20
DIAGNOSTICS Module Only	\$20
DISK MANAGER Module Only	\$5
DISK MANAGER 2 Module Only	\$10
HUNT THE WUMPUS Module + Book	\$8
OTHELLO Module + Book	\$8
PARSEC Module + Book	\$8
PERSONAL RECORD KEEPING Module + Book	\$8
TERMINAL EMULATOR II Module + Book	\$20
TI EXTENDED BASIC Module + Book	\$25
TI-WRITER WORD PROCESSOR Module + Book	\$25
TOUCH TYPING TUTOR Module + Book	\$8
TUNNELS OF DOOM Module + Book	\$12
VIDEO GRAPHIS Module + Book	\$8

Bye for now.

1. DISCLAIMER

The author and TISHUG Australia (Inc) can accept no responsibility for any damage which any reader may cause to any computer as a result of information contained in this article. Readers are advised not to try to measure any voltage which may exceed the interface ratings of the multimeter described. This article in no way endorses the Metex multimeter to be suitable for any measurement needs.

2. INTRODUCTION

Some time ago I purchased a METEX M-3650CR digital multimeter which has a normal digital LCD display, bar graph and a serial interface suitable for connection to the RS232 port of a computer. This allows the multimeter to perform front end analogue to serial data conversion and the computer to be used as a data logger.

The multimeter has opto isolation in its transmit and receive data connections. The computer needs to provide positive and negative potentials from the RS232 card to the output of the opto-isolator to permit data to be sent from the multimeter.

My concept of a data logger is to record the data, such as voltages, current or resistance regularly, say once each minute, or once each five minutes, etc, and place both the time and reading in a disk file for further processing. Additionally, the program could "watch out" for large changes in reading and log these with the time of each occurrence. Once a multimeter with a serial data interface is successfully interfaced in both hardware and software to a computer with a time of day clock, the program can be written or modified to suit each user's special requirements.

The interface described lends itself to a large screen display of the multimeter readings through the use of sprites or even larger characters, possibly using an assembly program. (Mike Slattery has done some work in the area of large character displays in assembly.)

The METEX multimeter produces 2 to 3 readings per second when the COMM mode is selected or will produce a reading on demand when a character is transmitted to the multimeter on the serial data line. The option of getting a reading on demand looks very good because it overcomes synchronism problems between the computer being ready and the meter sending the data. When the meter is allowed to produce its data automatically, problems occur with half strings, etc, because the computer sometimes only becomes ready to receive part way through a string.

A Terminal Emulator program, such as Fast Term, is not suitable for use as a data logger because of the inability to include access to a clock in the computer. When I tried to direct access the multimeter with Extended BASIC, I just ended up with error messages.

3. PROGRAMMING

I settled on an assembly routine to interface with the multimeter, linked from Extended BASIC. It would also be quite feasible to use pure assembly, but I prefer to write most of my programs in TI's powerful Extended BASIC and just the parts needed in assembly. This also provides an assembly program module for non-assembly programmers who are capable of programming in Extended BASIC.

The programs included in this article show a very simple Extended BASIC program which loads the assembly, opens the RS232 file and links to the assembly program. The second program is the source file listing for the assembly program. This file must be assembled into an object file which matches the naming in line 140 of the Extended BASIC program, ie, the object file is named "METEO" and it is located on disk 1.

The CALL LINK allows the specification of a return string variable, followed by row and column. The ROW is any integer value between 1 and 24. The COLUMN is any integer value between 1 and 32. The assembly program will print the reading on the screen if a ROW and COLUMN are specified. If they are not specified, ie, the CALL LINK just contains a string variable, eg, CALL LINK("MXREAD",METEX\$) then the assembly routine does not print on the screen and just returns the data as a 13 byte string in the variable METEX\$.

The assembly program sends a byte with the ascii value of "1" to the multimeter, to which the meter responds with a 14 byte string. The last character of the string returned is a Carriage Return (ascii 13) which is ignored. Once the whole string has been received in the assembly program, it is returned to Extended BASIC. After sending the byte to the meter, the program waits for 1.5 to 2 seconds. If no reply is received, the assembly program "prompts" the meter again. This system overcomes a "hang-up" condition which would occur if the meter was turned on after the program had sent the prompt byte. The assembly program also responds to FCTN 4 (CLEAR), allowing a return to Extended BASIC from assembly if FCTN 4 is pressed.

4. PHYSICAL INTERFACE

The interconnections between the TI99/4A RS232 and the METEX M-3650CR multimeter are as follows.

INTERCONNECTION CABLE.

METEX	TI99/4A
M3650CR	Computer
Serial	
Socket	RS232/1

LCD end
of socket

DTR	Blue	2	Serial Data IN, Rx from m/m
RTS	Green	6	DSR +12 volts via 1.8K ohm
		20	DTR Input
DX	Yellow	8	DCD Out, negative for Rx I/F
TX	Red	3	Serial Data OUT, Tx to m/m
GND	Brown	7	Signal ground

End of socket
nearest input
terminals.

The designations in the Metex book don't make a lot sense, but the above diagram shows how the connections must be set up to work and the wire designations from the Metex booklet. The "DX" is thought to be a misprint in the circuit diagram in the Metex booklet, which should possibly read RX.

Pin 20 of the RS232 should also be tied to pin 6 for the "side-car" RS232, the Multi-Function card and mini PE systems produced by Peter Schubert. The TI PE box RS232 card has an inbuilt pull up resistor for the pin 20 input.

The serial output circuit of the multimeter consists of a resistor and the transistor of an opto coupler. There are 3 wires. The Serial Data IN (pin 2) of the RS232 requires that the input data waveform is normally a negative voltage with respect to earth and the data must drive the input positive with respect to earth. There are two outputs from the TI PE box RS232 which may be used, pin 8, Data Carrier Detect (DCD) and pin 5, Clear To Send (CTS).

Originally my software used pin 5 to supply the negative interface voltage, but when it was tried with an old "side-car" RS232, it was found not to work. On the RS232 of the Multi-Function card developed by TIsHUG member Peter Schubert, pins 5 and 6 are wired together and via 1833 ohms of resistance to +12 volts DC supply. This only left pin 8 of the RS232 which was found to work OK with both the side-car RS232 and the PE box RS232 card. It is unknown at the time of writing if the software will work with other manufacturer's RS232s. If there are differences and if the CRU base and offset for RTSON are known, then modification of the source code by other users would be possible.

The RTS (DCD, pin 8) must be a positive voltage whilst transmitting from the RS232 and a negative whilst the RS232 is receiving to provide the required negative interface potential. The switching of the DCD caused a false first character of (ascii) zero on each string but this was fixed in software.

The positive side of the interface potential was derived from the Data Set Ready (DSR), pin 6, which is a 1.8K ohm resistor to +12 volts DC from within the RS232 card in the case of the PE box card.

Interface signals were measured as follows:

a) PE BOX RS232

RS232 pin 2, computer receives: -3 volts / +5 volts when pin 8 is giving -7 volts and +3.5 volts when pin 8 is at +8.5 volts. If data is sent to pin 2 by the meter while pin 8 is positive, it is not recognised by the RS232 card.

RS232 pin 3, computer sends: +/- 8 volts.

b) "SIDE-CAR" RS232

RS232 pin 2, computer receives: +/- 4 volts when pin 8 is producing -10 volts and +2 volts when pin 8 is at +8.5 volts. If pin 8 is at +8.5 volts and data is sent by the meter, it swings between +2 volts and +4 volts on pin 2 and it is not recognised the RS232 card.

RS232 pin 3, computer sends: +/- 10 volts.

The multimeter has a battery saver, which turns the meter off automatically if it appears unused for a period of time. The battery saver appears to be disabled by continual data access from the serial input.

5. EXTENDED BASIC TEST PROGRAM

```
100 ! SAVE DSK1.LOADMETEX
110 CALL CLEAR
120 DISPLAY AT(16,1):"Testing Metex"
130 CALL INIT
140 CALL LOAD("DSK1.METEO")
150 OPEN #1:"RS232.TW.DA=7.PA=N.BA=1200"
160 CALL LINK("MXREAD",METEX$,23,3)
170 DISPLAY AT(18,1):METEX$
180 GOTO 160
```

6. ASSEMBLY SOURCE FILE LISTING

* Source=METEX Object=METEO

* METEX 12th January 1993 2230 Ross Mudie.

* Register usage:

```
* R0 VSBW, NUMREF, STRASG
* R1 VSBW, NUMREF, STRASG
* R2 Pointer in BUFFER, STRASG
* R3 Length of string counter
* R4
```

```
* R5 Resend timer when no response after 1.5 to 2 secs
* R6 Screen start position from Row and Column
* R7 Flag for DISPLAY on/off
* R8 Anti lock up timer for comms interference
* R9
* R10
* R11 Return address
* R12 CRU base address
```

```
IDT ' METEX '
DEF MXREAD
```

```
RS2DSR EQU >1300
LED EQU 7 RS232 card LED when R12= >1300
RS2PT1 EQU >1340
RTSON EQU 16 RS232/1 pin 8 when R12= >1340
RIENB EQU 18 TMS 9902 Receiver Interrupt Enable
RBRL EQU 21 9902 Receive Buffer Register Loaded
XBRE EQU 22 9902 Transmit Buffer Register Empty
```

```
KSCAN EQU >201C
VSBW EQU >2020
STRASG EQU >2010
NUMREF EQU >200C
XMLLNK EQU >2018
CFI EQU >12B8
```

```
FAC EQU >834A
STATUS EQU >837C
KEYIN EQU >8375
GPLWS EQU >83E0
```

```
BUFFER BSS 16
SAVRTN BSS 2
WS BSS 32
```

```
B01 BYTE >01 Byte sized 1
FCTN4 BYTE >02 Fctn Clear from TI99/4A Keyboard
B13
```

```
CR BYTE >0D Carriage Return (Enter)
EVEN
```

```
D14 DATA 14 The length of a string from METEX
```

```
*-----
* From X/BASIC CALL LINK("MXREAD",RS$[,ROW,COLUMN])
```

```
MXREAD MOV R11,@SAVRTN
LWPI WS
CLR R7 If 1 argument to prevent display
```

```
CB @>8312,@B01 How many arguments in link?
JEQ ENDRC Don't get Row & column
```

```
GETRC INC R7 Set the flag to allow display
```

```
CLR R0 Element 0
LI R1,2 Argument 2, Row
BLWP @NUMREF Row as floating point number
BLWP @XMLLNK Convert FP number to integer
DATA CFI
```

```
MOV @FAC,R6 Store the integer row
DEC--R6- Decrement to start at 0
SLA R6,5 Multiply by 32 for VDP address
```

```
INC R1 Argument 3, column
BLWP @NUMREF Column as floating point number
BLWP @XMLLNK Convert FP number to integer
DATA CFI
```

```
A @FAC,R6 Add the column value to the row
DECF R6 Decrement for 32 COLUMN screen
```

ENDRC

* Start turning RS232 on here. RS232 was "opened" in X/B

```

WAIT LI R5,500 ] WAIT is a delay to allow received
LOOP3 DEC R5 ] data to finish when the program and
      JNE LOOP3 ] serial data have become out of sync
      JMP TXLOOP ] due to RS232 being plugged out/in
*                               ] on meter or when meter is turned on.
      END

```

```

LI R12,RS2DSR Set CRU Base for 1st RS232 card
SBO LED Turn LED on RS232 card on

TXLOOP LI R12,RS2PT1 Choose RS232/1 for CRU base
SBO RTSON RS232 pin 8 becomes POSITIVE

LDCR @B01,8 Load CRU with byte of 1 and send

LI R5,2000 Resend after this number of loops

SBZ RTSON Tx off, Rx on, NEGATIVE on pin 8

LI R2,BUFFER+1 Address to store received chars
CLR R3 Length of string counter

LI R8,500 ] Timer to prevent lock up due to
DEC R8 ] mm serial lead being plugged out/in
JEQ WAIT ] partway through character reception

TB RBRL Anything being received in Rx buffer?
JNE LOOP2 Loop if yes
SBZ RIENB Clear Rx buffer

```

*Test RS232 for characters & keyboard for FCTN 4 (CLEAR)

1. DISCLAIMER

DEMONSTRATION OF METEX DATA LOGGING ON THE TI99/4A
by Ross Mudie, 6th February 1993

Did you miss seeing the METEX multimeter communicating via the RS232 of a TI99/4A computer? It was performing a data logger function on the power mains voltage at the Sydney TISHUG meeting on 6th February. A good number of members saw the demonstration, but there were also quite a few who didn't see it. For all who missed out and those who want to have another look at the ideas, come along to the meeting on 6th of March at Ryde Infants school at 1.30pm.

There is an article describing interfacing the Metex multimeter elsewhere in this TND. This development can be used by extended BASIC programmers as the assembly language part has been developed. Members who saw the technique demonstrated at the February meeting identified potential to use the idea for a wide range of electronic and electrical recording functions. Some of the suggestions included monitoring electronic equipment which is failing intermittently and pool water pH. The logging files which may result from a measurement period can be further processed to provide statistical results or graphing programs could display results, either on screen or print out via the computer printer.

The demonstration will be set up upstairs, in the room used for the Assembly language class. Come and have a look at the ideas presented, your questions and interest will be welcome. This is an ideal use for your TI99/4A.

CHANGE OF DATE

for
TUTORIAL WEEKEND on EXTENDED BASIC
by Ross Mudie, 6th February 1993

Due to work commitments the proposed Extended BASIC weekend has been moved to Saturday and Sunday 17th and 18th April 1993. For all other details please see page 20 of the Jan/Feb 93 TND.

If you are interested in attending, please contact Warburton at (02) 918 8132 or Ross Mudie (02) 456 2122, on or before 6th March 1993. The decision to proceed with the tutorial weekend will be made at the meeting on 6th March 1993.

This weekend is for beginners and people who are having problems learning to use Extended BASIC. There is NO minimum standard, just a desire to learn and being prepared to make the effort.

```

TESTRS LI R12,RS2PT1 Tell CRU RS232/1

TB RBRL Character being received?
JEQ GETRSC Yes, full char in Rx buffer

BLWP @KSCAN Scan keyboard
CB @FCTN4,@KEYIN Is it FCTN 4 (Clear)
JEQ EXIT Jump if yes

DEC R5 Is R5 down to zero yet?
JEQ TXLOOP Resend when R5 is zero
JMP TESTRS Else back around the loop

GETRSC STCR R1,7 Store assembled char on reg 1
SBZ RIENB Disable Rec interrupt, reset RBRL

INC R3 Current character counter
C R3,@D14 Has counter reached 14?
JLT SARXCH Not yet, save current character
CB R1,@CR 14th character recd, is it a CR?
JEQ EXIT If yes, all is OK, send to X/B
JMP WAIT NO, Goto WAIT, allow Rx data to end

SARXCH MOV B R1,*R2+ Save received character in BUFFER

MOV R7,R7 Test DISPLAY flag
JEQ TESTRS Jump if display is off

AI R1,>6000 Extd basic screen offset
MOV R6,R0 Row and Column VSBW base address
A R3,R0 Add position in input str FOR VSBW
BLWP @VSBW Display on screen
JMP TESTRS Go and get the next character

EXIT LI R12,RS2PT1
SBO RTSON RS232 pin 8 becomes POSITIVE

LI R12,RS2DSR
SBZ LED Turn LED on RS232 card OFF

MOV B @B13,@BUFFER Put string length in BUFFER+0

CLR R0 Element 0
LI R1,1 Argument 1 in CALL LINK
LI R2,BUFFER Assembly Address of the string
BLWP @STRASC Send string to x/b

CLR R0 ]
MOV B @STATUS ] Normal method of returning
LWPI GPLWS ] to the calling program.
MOV @SAVRTN,R11 ]
B *R11 ]

```

THE ART OF TROUBLESHOOTING
by B.v.Takach

The title I have given to this contribution is a misnomer. Troubleshooting is not an art, it is a systematic logical progression from the source to the target.

Having thus specified it, I should be forgiven for the title, every reader will agree it looks a lot better than its precise definition. Let me illustrate the process of troubleshooting with a case history.

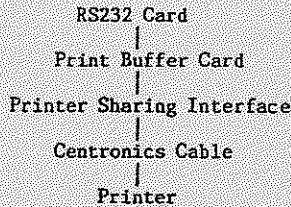
I was preparing a number of title pages for a folder. Each A4 page was supposed to have a single line centred on the 20th line of the page. First, I typed in the titles one after the other using TI-Writer with the appropriate left margin. Then I fed the first page into the printer and set the T-offset to the 20th row.

The first title happened to be on line 179, so the FCN9, PF, 179 179 PIO command sequence produced the first perfect page. Pressing the form feed button promptly lined up the next page, which was completed using FCN9, PF, 180 180 PIO command sequence. And so it went page after page until I typed in FCN9, PF, 187 187 PIO. This time nothing happened!

I repeated the command several times without the slightest sign of any activity from the printer. Now the troubleshooting process had started in earnest.

- Printer on line, light on;
- Manual printer functions OK;
- Switch off printer, restart, repeat print command. No joy;

Consider the print path from PE box to printer:

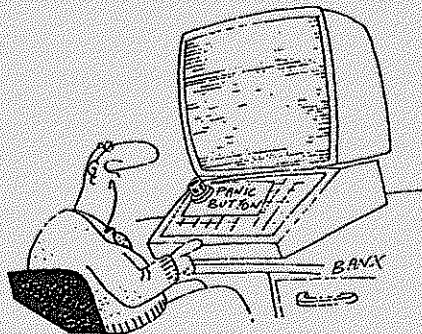


Repeated the print command and observed the led indicators of the 32K exp.mem, the RS232 and the print buffer cards - all three leds gave a quick wink. Hunted for a cable to connect the RS232 card direct to the printer, and repeated the print command - no joy.

Now it became obvious that I was faced with an extremely complex problem: logically I had reached the end of the road. Meanwhile the screen went blue and I had the blues.

It was then I decided to examine line 187. Line 187 was a blank line!

So, once more the problem was solved using faultless watertight logic. No story should end without a moral, and neither should this one. The moral to the story - as I said at the beginning - follow the path from the source to the target, not from the target to the source!



GIF FILES ON THE BBS
by Ross Mudie

GIF Graphics files are usually in Display Fixed 128 format, which are not suitable for download or upload on the TEXPAC BBS. To overcome this problem, a small number of GIF files have been implanted into Extended Basic program format as a trial. These special programs are prefixed G_ in the program menu. Download and save these as Extended Basic programs as per normal. To create a D/F128 GIF file from one of these programs, just RUN the program. Every thing is contained in the program which will write the required GIF file to disk when it is run. There will be an article written on the technique used in a few days of this article being placed on the BBS.

The size of Extended Basic programs via the BBS has a practical limit of 45 sectors (to avoid using CALL FILES). The first of the GIF files will be those of less than 44 sectors in the original D/F128 format. Larger GIF files are implanted in "chained" extended basic programs. These will have the prefix G_ and the suffix 1XB for the first part, 2XB for the second part, etc. It will be necessary to download all parts of a particular implanted file and then RUN the one with the suffix 1XB. It will then chain to the 2XB part, then the 3XB part and create one big D/F128 GIF file.

This is a trial, please let me know how you get on. To view your newly created GIF files, use GIF-MANIA which may also be downloaded from the program menu, since it is also implanted.

Please let me know if you will use the BBS as a way of getting GIF files, if they become available this way in the future. Mail to SYSOP is the easiest way to reply when on the BBS.

CONTINUED FROM PAGE 11

```

*****
410 MAXREADING=MAX(READING,MAXREADING)
420 IF FIRSTFLAG THEN 430 ELSE FIRSTFLAG-1 :: MINREADING
    =READING
430 MINREADING=MIN(READING,MINREADING)
440 GOSUB 580 :: IF OLDMINUTE=MINUTE THEN 370
450 OLDMINUTE=MINUTE :: CALL SOUND(100,900,0)
460 AVG=INT((RDT/R)*10+.5)/10 ! 1 Decimal place
470 DISPLAY :TIME$;R;TAB(9);AVG;TAB(16);MINREADING;TAB(2
    1);MAXREADING
480 IF DISKLOGON THEN PRINT #2:TIME$;R;TAB(9);AVG;TAB(16
    );MINREADING;TAB(21);MAXREADING
490 MINREADING,MAXREADING,FIRSTFLAG,R,RDT=0
500 GOTO 370
510 ! Date subroutine
520 CALL PEEK(-31158,CDAY,CDAY,DATE,DATE,MONTH)
530 D=DATE :: DATE=DECIMAL
540 D=MONTH :: MONTH=DECIMAL
550 MONTH$=STR$(MONTH):: IF LEN(MONTH$)=1 THEN MONTH$="0
    "&MONTH$
560 DATE$=STR$(DATE)&"/"&MONTH$&YEAR$
570 RETURN
580 ! Time Subroutine
590 CALL PEEK(-31164,SECOND,SECOND,MINUTE,MINUTE,HOURL)
600 D=HOUR :: HOUR=DECIMAL
610 D=MINUTE :: MINUTE=DECIMAL
620 D=SECOND :: SECOND=DECIMAL
630 HOUR$=STR$(HOUR):: IF LEN(HOUR$)=1 THEN HOUR$="0"&HO
    UR$
640 MINUTE$=STR$(MINUTE):: IF LEN(MINUTE$)=1 THEN MINUTE
    $="0"&MINUTE$
650 SECOND$=STR$(SECOND):: IF LEN(SECOND$)=1 THEN SECOND
    $="0"&SECOND$
660 TIME$=HOUR$&MINUTE$ &" ":"&SECOND$
670 RETURN
680 CLOSE #1
690 IF DISKLOGON THEN CLOSE #2 :: DISPLAY : "Disk file " ;
    DISKLOGNAME$:"has been closed"
*****

```


Regional Group Reports

Meeting Summary For MARCH

Banana Coast 14/03/93 Sawtell
 Central Coast 13/03/93 Saratoga
 Glebe 11/03/93 Glebe
 Hunter Valley 13/03/93
 Illawarra 15/03/93 Keiraville
 Liverpool 12/03/93 Yagoona West
 Northern Suburbs 25/03/93
 Sutherland 19/03/93 Jannali

BANANA COAST Regional Group
 (Coffs Harbour Environs)

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

CENTRAL COAST Regional Group

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

GLEBE Regional Group

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

HUNTER VALLEY Regional Group

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

ILLAWARRA Regional Group

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

LIVERPOOL Regional Group

Regular meeting date is the Friday following the Tishug Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 644-7377 (home). After 9.30 PM or at work (02) 708-1987 Liquorland Yagoona for more information.

12TH FEB 1993 *****
 My Place * TI ARTIST AND *
 34 Colechin St * PAGE PRO *
 Yagoona West 2199 *****

MARCH 1993 - NO MEETING

APRIL (Good Friday) *****
 My Place * Should be some *
 34 Colechin St * new programs *
 Yagoona West 2199 * from overseas *

NORTHERN SUBURBS Regional Group

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

SUTHERLAND Regional Group

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

TISHUG in Sydney

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

MARCH MEETING - 6th MARCH

There will be another demonstration of the brilliant idea that Ross Mudie put together- the interfacing of a multimeter to the TI to monitor voltages, resistances etc of such things as the mains output. Any new software will be examined.

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

April 7th March
 May 11th April

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

FOR SALE

- * disk drive stand alone plus transformer \$ 20.00
- * Peter Schubert modem \$ 20.00
- * assorted cables (each) \$ 2.00
- * multi-module plug in holder \$ 10.00
- * complete Editor/Assembler \$ 25.00
- * cartridge and manual "Home Budget" \$ 10.00
- * complete TI-Writer Word Processor \$ 20.00
- * TI Extended Basic module and book \$ 20.00
- * Mini Memory module and book \$ 20.00
- * "Plato" All School Subject Lessons on computer disk, year 3 to 12. 110 disks- \$ 2.00 each, plus module, catalogue disk and survey disks for \$25.00. \$245.00
- * "SAMS" Graphics and sound cassettes and manual \$ 7.00
- * "SAMS" Basic Programs and cassettes and manual \$ 7.00
- * "SAMS" Basic Tricks for TI-99/4A, cassette and manual \$ 7.00
- * cartridge "Terminal Emulator 2" \$ 10.00
- * cartridge "Disk Manager 2" \$ 10.00
- * Introduction to Assembly Language for the TI Home Computer \$ 7.00
- * TI Home Computer Technical Data Manual \$ 6.00
- * TMS 9900 Microprocessor Data Manual \$ 6.00
- * 101 Programming Tips for the TI-99/4A \$ 6.00
- * 36 TI Programs for Home, School and Office \$ 6.00
- * Programmer's Reference Guide to the TI-99/4A \$ 6.00
- * I Speak Basic to my TI-99/4A \$ 6.00
- * TI Basic Computer Programs for Home \$ 6.00
- * 33 Programs for the TI-99/4A \$ 6.00
- * TI Games \$ 6.00
- * Stimulating Simulations for the TI-99/4A \$ 6.00
- * Advanced Diagnostics \$ 6.00
- * 99/4A Auto Spell-Checker \$ 8.00
- * Flight Simulator for the TI-99/4A \$ 5.00

Klaus Umland
 8 Belloc Pl.
 Winston Hills 2153
 Tel. 639 7179