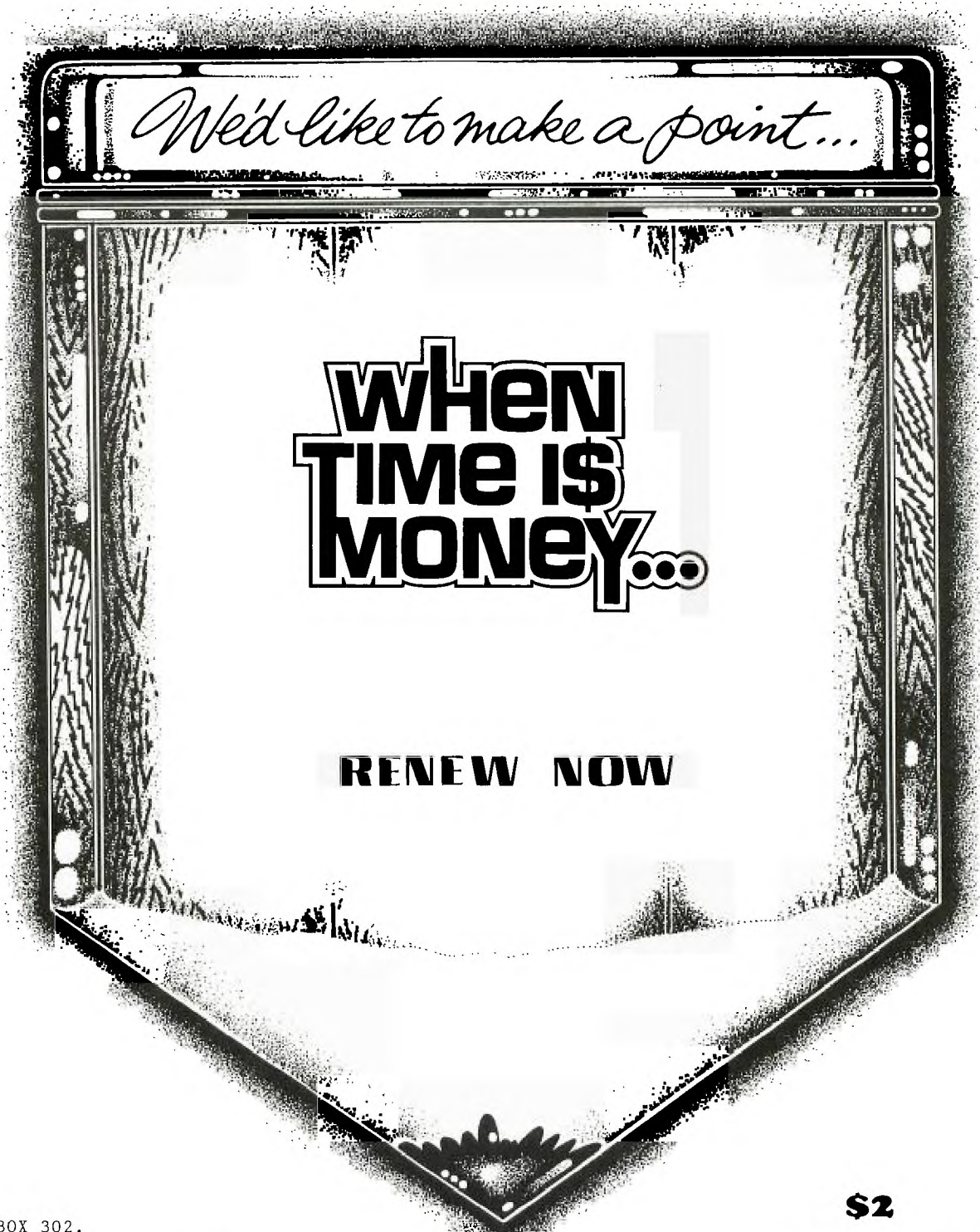


④42 8703
T.I. S.H. U.S.

REGISTERED BY AUSTRALIA POST PUBLICATION No. NBH5933

T SHUG **NEWS DIGEST**



We'd like to make a point...

**when
TIME IS
MONEY...**

RENEW NOW

P.O. BOX 302,
CARLINGFORD,
New South Wales, Aust. 2118

\$2

MARCH 1987 (Vol 6. No. 2)

TI99/4A Owners Home Computer
User Group
TISHUG NEWS DIGEST

MARCH 1987

Correspondence to:

PO Box 302
CARLINGFORD NSW 2118

Texpac BBS: Tel.: (02)319.1009

COMMITTEE MEMBERS:

Co-Ordinator:

Chris Buttner..Tel.(02)8717753

Secretary:

Terry Phillips.Tel.(02)7976313

Treasurer:

Bert Thomas....Tel.(047)541535

Publications:

Bob Montgomery.Tel.(042)286463

Sysop:

Ross Mudie.....Tel.(02)4562122

Merchandise:

Cyril Bohlsen..Tel.(02)6395847

Technical:

John Paine.....Tel.(02)6256318

Librarian:

Terry Phillips.Tel.(02)7976313

REGIONAL COMMITTEE MEMBERS:

Glebe:

Mike Slattery..Tel.(02)6920559

Penrith:

John Paine.....Tel.(02)6256318

Central Coast:

Russell Welham.Tel.(043)924000

Liverpool:

Stan Puckle....Tel.(046)256157

Illawarra:

Rolf Schreiber.Tel.(042)842980

Bankstown:

Peter Pederson.Tel.(02)772396

Carlingford:

Chris Buttner..Tel.(02)8717753

Sutherland:

Peter Young....Tel.(02)5288775

Manly Warringah:

Dennis Norman..Tel.(02)4523920

Coffs Harbour:

Keir Wells.....Tel.(066)551487

MEMBERSHIP AND SUBSCRIPTIONS:

Joining Fee.....\$ 8.00

Annual Family Dues.....\$25.00

Dues O'seas Airmail...US\$30.00

Publications Library....\$ 5.00

Texpac BBS.....\$ 5.00

BBS Membership:

Other TI User Group

Members.....\$10.00

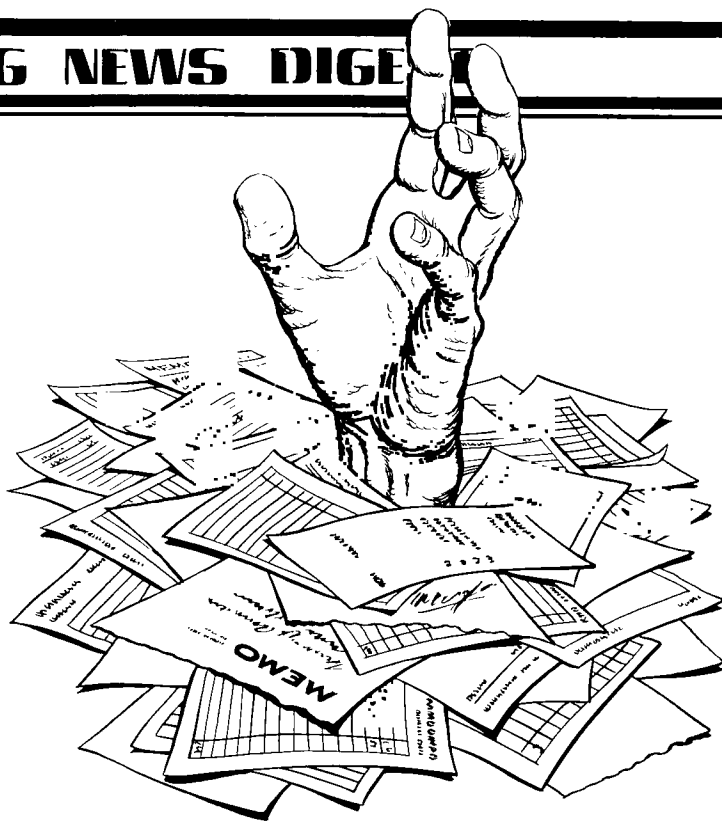
Public Access.....\$25.00

GROUP GENERAL MEETING:

First Saturday of each Month at
Woodstock Community Centre,
Church Street Burwood. Starts 2pm

COMMITTEE MEETINGS:

Held immediately after the
General Meeting at the same
venue.



CONTENTS

Page 2 - General Information and editorial

Page 3 - AGM Report by Chris Buttner

- IBM Compatibility

Page 4 - Techo Time with John Paine

- New Secretary Terry

- The Communicators by Ross Mudie

Page 5 - Techo Time cont.

- RS232 Interface Box with 32K Memory

by Peter Schubert

Page 6 - Software with Terry

Page 7 - XB Screen Colors from Tacoma 99ER's

Page 8 - Programs

Page 9 - Programs

Page 10 - Younger Set with Jenny

Page 11 - TI Shop

- Super Widget

Page 12 - Link It with Ross Mudie

Page 13 - A Look at Speed by R.A. Green

Page 14 - A Look at GPLLNK by R.A. Green

Page 15 - Forti Music System by Arto Heino

Page 16 - Multicolor Mode by Stephen Peacock

page 19 - TI Writer Spricht Auch Deutsch

by Ben Takach

Page 20 - Regional Reports

This edition of the TND has been put together Geoff Trott, Rolf Schreiber, Terry Phillips, Brian Graham under the guidance of Shane Andersen. The new editor almost missed the paste up after having been given the wrong address of Shane's place. Thanks to Shane for allowing the paste up at his place.

A big thank you should be given to Shane for the amount of time that he has spent in editing for the past six years. Here's hoping he enjoys his retirement.

Future magazines will be done in Wollongong with the Illawarra Regional Group assisting.

This issue has a lift out for renewal purposes and should be used when paying your dues. There are the regular features that have appeared in previous issues as well as a number of assembler programs.

Finally, I have taken on this position of TND Editor to assist the group to achieve its aims. I am looking forward to at least maintain the current standard of the magazine. This will only be done if many members contribute to its content.

Bob Montgomery

AGM REPORT by CHRIS

The AGM is over for another year and quite a few changes made - in personnel and format. Two life-memberships were created: one for John Robinson and the other for Terry Phillips in recognition of the years of service given to the club. Congratulations! To those members who retired from the committee - thank you for your dedication and the personal sacrifices made for the benefit of the club: to the new committee members - with your support I hope we can all have a fulfilling year in TISHUG.

As club members, we should be able to speak out frankly and sincerely to one another sharing our triumphs and failures, high times and problems. In that spirit, I want to share the following thoughts and ideas with you.

There is an ever present tendency to take for granted what happens around us and the club environment is no exception. For members of this club, technical information and assistance has been available freely for a number of years. The same is not true of all other computer user groups. Recently, I was astounded to discover the technical officer of one such group would not provide information to a member other than to say read the manuals. Whatever the rights or wrongs of this approach, it does serve to highlight the level of support available to all members of this group. The service is there for the asking either direct from the technical co-ordinator or regional groups. Use the services and facilities provided and share your experiences with others to help expand the fellowship which we have through ownership of "the orphan".

Your newly elected committee met on Tuesday 10th February and I am pleased to say, took the bit between its teeth and is off and running. In a very short space of time you should see major improvements in club services, the first of which is likely to be the BBS. Hopefully, by the time you read this, it will be operating at 300/300, 1200/75 and 1200/1200. For those country members on the BBS this should substantially lower your phone bills. If you are calling STD, remember the cheapest method is to pull off the information you want, saving it to disk. You can then browse at your leisure and log back on if you need to send any messages.

Having now mentioned the committee let me EMPHASISE we are only human. Some things are possible, others ... Many of you I fear, voted with your hearts and not your heads at the AGM. I have no doubt the intent was to create a more "democratic" and hopefully more responsive committee. Let me now summarise for you the results of the voting. Four committee members are entitled to 2 votes each: there is a real problem in determining what a quorum should be since the number of committee members is not fixed: there is a further consideration attaching to regional groups - how many members must there be in the "group" before it is acknowledged as a regional group and is accorded voting rights at committee meetings. That's just for starters!

What's the bottom line? The changes brought about by your voting are so far reaching that the memorandum and articles of association prepared so generously for us by Brian Graham will need to be completely rewritten. Brian is unable to assist in this task so the job will have to be done on a commercial basis and that doesn't come cheaply; not to mention the delay in effecting incorporation which will necessarily follow on from that. It's not a headache, it's an absolute nightmare!

Those few topics should give you some idea of the magnitude of the problem confronting the committee. All this and the bread and butter running of the club you expect us to carry out in our spare time. I am sure I speak for the other members of the committee when I say we will give it our best shot but please respect our humanity!

If there is any lesson in all this it surely is: be realistic in what you expect committee members to do - share the work around - make sure the committee has an easily managed framework and lastly don't duck the issues and challenges in running your own club by nominating regional groups or some other form of committee. The club is for PEOPLE. Lets all work together to keep it that way.

Your committee comprises mainly new members and to them I extend a hearty welcome as together we accept the challenges which lie ahead. I am sure you will see new things happening in the club. If I have said something you don't agree with speak up. There is every chance your point of view will appear in the TND. If you want to write or send E-Mail, I can be contacted through the club P.O. Box or CO-ORD on the BBS. The next committee meeting will be held immediately following the March monthly meeting.



IBM COMPATIBILITY FOR THE 99/4A

This is a two part system comprising a small bridge box that connects to the I/O port on the 99/4A and a Turbo XT capable of running at 4.77Mhz and 8Mhz.

The XT unit houses power supply, motherboard, colour graphics card (RGB and composite), Floppy Disk controller, 1 half height DS/DD disk drive, parallel port and 256K Ram (with sockets to upgrade to 640K). There are 8 expansion slots (2 used by Disk Controller and CGA card.)

The bridge box contains the inputs for Video and outputs for keyboard and monitor, as well as software to switch between the 99/4A and the XT and also convert the 4A keystrokes into XT key codes. The PE-Box can be connected to the bridge box.

The only items shared by the two systems are keyboard and Monitor/TV. 40 and 80 column modes are available. It is possible to have the XT and 4A running concurrently. Any IBM cards of your choice can be added to the XT system since it has IBM software and hardware compatibility.

Minimum requirements are a TI 99/4A console and a monitor or TV with modulator.

Marketing is being done by Triton Products Company in San Francisco, CA. and the system comes with a 30 DAY money back guarantee and 1 year parts and labour warranty. The cost - US\$499 plus shipping and handling. Deliveries are planned to commence on 1st March.

Note: The above information is an extract from an article which appeared on the GENIE BBS. There is no specific mention of the designer/s but an educated guess suggests this is the Millers Graphics unit.

TECHO TIME



It appears that a couple of oversights had occurred in the Techo Time column on page 20 of the September '86 newsdigest.

The problems are as follows. The undesignated fet-like components in the composite video circuit are actually bi-lateral switches (4016 or 4066). The other fault is that the composite video output needs some sort of positive biasing to function correctly.

The time has come the Walrus said to speak of many things

As of the February '87 Annual General Meeting and this issue of the newsdigest I have been replaced as the Technical Co-ordinator of our club (and I believe better served) by John Paine. At this time I believe that the Techo Time column and BBS user name will be continued. John should be able to be contacted through both mediums, or by any other means as specified by John.

REMEMBER, John can't think of everything or do everything by himself, so give him all the assistance you can.

Robert (former Tech. Co-od.)

MEET THE NEW TECHNICAL CO-ORDINATOR - JOHN PAINE



Techo Time

Continued page 5

THE COMMUNICATORS, TEXPAC BBS NEWS. By Ross Mudie, TEXPAC SYSOP, March TND.

After the foretaste of the SYSOP job from September 1986, I have been elected to the BBS SYSOP position for 1987. The BBS is now using a much modified TEXPAC code which runs in TI basic with quite a lot of linked assembly. This software utilises the available TI99/4A hardware in what I believe is the most reliable and practical manner. The BBS operated without on-site SYSOP intervention from 24/12/86 to 2/2/87 using the remote SYSOP facilities. (Users should note that when the remote SYSOP facilities are in use the BBS will not answer calls on the normal number.)

The TISHUG committee has purchased a smart modem which will permit operation on 300/300, 1200/75 or 1200/1200 bauds. This, coupled with the upgraded software which reads the disk files in assembly and "prints" to the RS232 in the same assembly program, will allow users to download files at better than 4 times the old rate if 1200 bauds is used. This will reduce STD phone charges for out of towners and allow more information to be obtained per session. Regular users of the BBS will already have noticed the difference on 300 baud operation.

A program to send mail from a DISplay VARIABLE 80 file to other BBS users is available on the BBS. There is a full description of how to send mail on the BBS in the file named SENDMAIL in the NEWS menu of the BBS this month. The SENDMAIL program is in the downloadable software section. This allows any mail to be prepared and corrected using an editor then despatched to the bulletin board with maximum efficiency.

The MBP clock card in the BBS is to be replaced with a Triple Tech board. This allows further development and off site debugging of the programs for the BBS. The Triple Tech printer buffer will speed up the time taken for the BBS operation to be logged, reducing even further delays in operation.

New SECRETARY



Terry

This is my first column as Secretary, and there being little news between the AGM elections held on Saturday 7 February and the time I am writing this, then I don't have much to tell you at this time apart from one item that I see as very important.

Come the end of April it is renewal time for the vast majority of members. As you would have heard Fred Morris say at the AGM, our membership has dropped to a total of 451. In other words the heady days of a 1000+ membership are behind us and we must look to holding onto our current membership. I urge you then to make the commitment today and renew your membership. It is only with a strong, committed group that TISHUG will survive into 1988. Elsewhere in this issue is a convenient form for you to fill in and renew. As said earlier - do it now.

By the way the address for all correspondence is still PO Box 302 Carlingford NSW 2118. I am currently searching for a new post office box nearer to my work. When found you will be notified of the new address.

Bulletin Board members may send electronic mail to me - username : ARY.

Another new feature of the BBS is the optional Log-On-Message. This can be loaded remotely by the SYSOP and gives a short, important message to each user after the correct user number and password has been entered. As with most operations of the BBS, the Log-On-Message can be paused with <ctrl>S or exited from after the current line has been printed, by pressing E once only.

In response to the request of a BBS user, the BBS now tells who the previous user was and allows the user log to be examined.

TISHUG members can join the BBS by sending a \$5 annual subscription (additional to standard membership) to the SECRETARY of TISHUG at PO Box 302, Carlingford. Non-members of TISHUG who are members of another TI computer user group may join the BBS for \$10 per annum, whilst the fee for a person with no affiliation is \$25 per annum. An \$8 joining fee applies to new non-TISHUG members.

The downloadable program section for March includes 22 different programs, 6 of which are assembly programs which load from extended basic, if the computer has a 32K memory expansion. Other basic and extended basic programs include music, mindbending games & utilities.

In common with the TND editor, the BBS SYSOP is continually in need of good material for use on the BBS and in the magazine. Material can be placed on the BBS as mail for both the SYSOP and EDITOR. This allows the material to be read off the BBS at a convenient time. The key to the effectiveness of TISHUG's purpose of real support for our "orphan" is ACTIVE PARTICIPATION by a large number of members, not just the committee.

The number for TEXPAC BBS is (02) 319 1009. The BBS is available 24 hours per day, 7 days per week. There is a session time limit of 30 minutes between 6am and midnight whilst from midnight to 6am the session limit is 60 minutes. Users are asked to allow 30 minutes between sessions to allow others to get on. Multiple program down loads are permitted within session limits.

TECHOTIME with John Paine

Salutations, fellow 99/4a's.

For those of you, that were unable to attend the Annual General Meeting in February, I wish to introduce myself as the newly elected "TECHO" of this active group of TEXAS INSTRUMENT HOME COMPUTERS Users.

My name, as the heading above implies is John Paine and I live in the far Western Suburbs of Sydney, at Mt. Druiitt. I have been a member of our group for just under two years so by some standards I may be considered a relative newcomer to the HOME COMPUTER USERS GROUP. So be it, although I may not have been seen to be very active by the majority I have been playing with our favorite computer for nearly four years and with the aid of other dedicated users I have endeavoured to learn, cram, steal, borrow as much knowledge of the innards and strange workings of this machine as possible and I now believe that we as a group are in the position to make our selves self sufficient in terms of hardware and diagnostic support which allows us to confidently persue our hobby with out the dreaded fear of writing off our financial outlays in equipment when and if our little machines have the occasional hiccup. (just think of how many manhours of pounding that poor little keyboard you and your favorite plaything have gone through together. Good grief, it is part of the family now and I believe that at last we can say if things are not well....Lets get the doctor, with out having to consult the SPECIALIST that resides in North Ryde. Think of your membership of TISHUG as your Medicare Card.)

So much for the soapbox routine, I see my responsibilities as the Resident "TECHO" as many and varied. I would like to take this opportunity to air my views on what you, the members may expect in the in the future.

- 1) Hardware Support in terms of an alternative Repair facility for consoles, PEB cards and bits and peices.
- 2) Research and Development of new products with production done in Australia. (The US Dollar and German Mark are so expensive.)
- 3) The ability to acquire some of the more common spare parts, eg. Keyboards, Power supply boards and module connectors from the club shop at reasonable rates.
- 4) Tips and suggestions to some common failure avoidance technicques.
- 5) The ability to implement new technology in peripherals without resorting to paying excessive exchange rate fluctuations.

In view of the comments above I would welcome your advice on "WHAT DO I WANT" and I invite you, The USER to indicated to "TECHOTIME" c/- of the Club PostBox or the BBS to User Name "TECHOTIME", to tell all your expectations.

Finally, at the AGM I announced the forthcoming availability of a locally developed Battery Backed Solid State Floppy Drive Emulator, (Ramdisk).

I can now state that the bare board will be available from the club shop at \$35.00 each. The functionally equivalent board to import from the US is \$50.00 US and subject to 25% import duty and if you are unlucky at time of arrival a further 20% Sales tax. (ouch). For the mathematically inclined you will find very little change from \$100.00.

Once again to introduce this product and refresh other memories, the TISHUG ramdisk is a Battery backed 192k byte storage device that thinks it is a double sided diskdrive which lives quite happily in the PE Box but allows access of files and programs at (almost) lightning speed.

The major beneficiary of such a device would be the owner of a single disk system, but all users will find this device more than usefull.

Unfortunately deadline time for this edition of the Magazine precludes too much detail for now but suffice to say that the initial production run will be only 50 boards so I suggest that you contact the shop with your orders and deposits.

The boards will be available for collection in early April.

One final point, these boards will have gold plated contacts, throughhole plating and a solder mask for reliable LONG TERM usage, not like some imported cards with which production costs were minimised with long term reliability compromised.

I may be contacted by mail at the normal club address shown elsewhere in this edition or if you wish to phone at reasonable hours i.e. not after 11.00 pm EST. at home on domestic 02-6256318
int.nat. 612-6256318
or during normal working hours at 02-8197200.

Cheers for now and lets see if our Want List is achievable. It is up to YOU.

RS232 INTERFACE BOX WITH 32K MEMORY

- # EXPANDABLE
- # THRU CONNECT FOR PE BOX, ETC
- # SHIELDED AGAINST RADIATION
- # 4 RS232 PORTS POSSIBLE
- # NO POWER CORD
- # 32K MEMORY PROVISION
- # 19200 BAUD MAX SPEED
- # 38400 BAUD FOR MIDI OPTION
- # BASIC UNIT ONLY \$99

This new expansion box in its basic form is similar to the original TI Stand-alone RS232, only it is up-to-date with the latest technology. The old TI box was large, used 3 circuit boards and a large power supply. This new design from TISHUG member Peter Schubert has only one main board which includes the TI Buss thru connection, and also provision for 32K memory expansion, fits into a box measuring 170x120x55mm, but best of all it uses no separate power supply, only power from your console. There is no power cord and it uses less power than a speech synthesizer.

All of TI's RS232 commands are supported (BA=, DA, PA, etc) Speed 300 to 19200 Baud (NOTE 110 baud has been deleted). If you already have a RS232 card, this unit can be used for ports 3 and 4. It is supplied ready to plug in and tested with one port for \$99 without box. Cutting and drilling details are supplied so you can make your own box, or Peter will fit it into an Aluminium diecast box for you, paint sprayed in Beige for \$30. A second RS232 port can be fitted with extra connector on back for \$45, or without RS232 conversion (for experimenting with 5v TTL interface) for \$30. 32K Memory fitted costs \$35 extra. This is the latest design using a single CMOS 32K chip and 3 small IC's.

If this all sounds good to you dont hold your breath, for there is more to come. This main circuit board design is called the 'Mini TI Expansion- Part One'. On each end of the board where the connectors are there is a row of pins extending up from the board onto which can be plugged another circuit board extending the interface to this board also.

The first such board that will be designed to plug into it is a very powerful DSDD Disk Controller. Another planned add-on will be a Parallel Port. The basic RS232 board with second Port added has available direct access for experimenting with your own gadget, such as RTTY interface, Robot control, Remote control, Alarm systems, Digitizer drawing pad, Midi interface, Mouse interface, etc, to mention just a few ideas that come to mind.

The Mini Expansion Box RS232 can be ordered from:
Peter Schubert
P.O.Box 28
Kings Cross, 2011
Phone (02)358 5602
PETESAKE on TEXPAC BBS

Postage for the unit without box is \$3 or \$6 fitted into box. It is available to Regional Group leaders for meetings if they can arrange to pick it up from Peter. ●



Well the AGM went for so long last meeting that time did not permit a proper shop set-up. We had to vacate the club premises by 5pm to enable them to set up for the evenings entertainment. Those of you who went to the car park may have got your software and other goods but judging on the length of the line to buy you would have been in for a long wait. If you couldn't wait or otherwise missed out have no fears. The February software releases will be available at the March meeting or if you were a visitor from the country especially for the AGM then you can order by mail.

My thanks to all the members who voted for me as Librarian. It will be my pleasure to again help you out with your software needs this year. Not forgetting the publications side of things, because for the first time these two areas are combined as one. Russell Welham, who has almost been in this group as long as me, has graciously accepted the job as my assistant and will look after the publications side of things. Although Russell lives up on the Central Coast I don't see this as any problem as, if I recall correctly, he has only missed about 1 Saturday meeting since joining TISHUG. There was some discussion at the AGM on streamlining software distribution. Members can rest assured that I have taken this on board and shall look into it to see what might be able to be done. One thing I don't want to do at this stage, until back-ups are available, is let a lot of software go to various Regional Groups. This is however, certainly an option that will be explored. It should be remembered that in the software library, while we have quantity, some programs are of low quality and for this reason I have deliberately avoided issuing some programs. Also in the library is a great deal of commercial software, and again, for this reason it has not been issued.

Enough of that for the time being. Now to what will be released at the Shop in March.

1. PROGRAMS FROM TITRONIC DATA - I had never heard of Titronic Data until last meeting when Tony Imbruglia handed me a disk of games and utilities he wanted released as soon as possible. Tony, himself is Titronic Data. Some of you may have a copy of one of his earlier efforts - Torpedo Attack - which was sold at the shop under a licencing arrangement a couple of years back. On this disk -XB and 32K expansion required- you get the following:

COMPLETE CATALOG - a cataloguing utility which will store and sort up to 500 filenames and give you a hard copy printout. It also enables you to print disk catalogs to place on disk jackets. The HELP file with this one contains some very interesting reading which will be beneficial particularly to novice disk users.

HAUNTED HOUSE - as the name might suggest this is a game where you have to move from room to room avoiding the ghosts that roam about. Your object is also to not only avoid the ghosts but to pick up the gold that is scattered throughout the rooms.

PRINT A CATALOG - this program does just that. Either to the screen or to your printer.

TI TARGETS - this is an arrow shooting game where you have to hit the moving target which goes at various speeds down the right hand side of the screen. You have 30 arrows to try your luck.

TORPEDO ATTACK - previously mentioned and some of you may have this one. It is still good fun to play and of course the object is to sink the enemy submarines while avoiding their torpedos. A dive bombing plane adds to the fun of this game.

WHO DUNIT? - this is a very well done solve the crime game. You have to guess who dunit, how they dunit and where in was done (dun!). Some good graphics and colorful screens make this a very attractive game to play.

Also on the disk as a load/menu program which is very well done.

Thanks for these Tony. I hope the members like it as much as I do and they encourage you to write more programs.

2. DISKNAME - MARCH1987 - contents:

SOME GUYS LIKE BEER - a game requiring XB and 32K expansion. There are no instructions in the game but here is how you play. Move your man with the number 1 joystick and walk over the barrel segments. As you do this they drop to the next lower level. The idea is to make up complete barrels. Naturally the "baddies" try to stop you. Sound Familiar? Right. If you have played Burgertime then this games objective is taken directly from it.

HOUSTON - a big musical hit from a few years back. This version is programmed by Bill Knecht, a very competent musical programmer from the USA. Some good graphics and a nice melody make this a pleasure to watch and listen. XB and 32K required.

IF I FELL - again a very big song hit from the Beatles. This version is excellently programmed with the words on the screen so you can sing along with the music. XB and 32K

SICILY 1943 - refight the World War 2 battle of Sicily. Full instructions are included in this very comprehensive game. Programmed in XB it occupies 78 disk sectors, so naturally you will need 32K expansion.

THE HAUNTED MINE - written by John Behnke, the author of several Tunnels of Doom Data Bases. This one follows on that theme. All adventurers will get a great kick out of this game. Full instructions are included, and like all others on this disk it runs in XB with 32K expansion.

3. JOHN TAYLOR'S SPRITE EDITOR - more good software from John. This is one of the best sprite editors I have had the pleasure of using. Full documentation is on the disk and, as a bonus, it is released as a "flippy" with 127 pre-defined sprites on the flip-side. It is very well done and this disk should not be missed.

4. MUSICPREXN - this disk is not for everybody, but if you are interested in music and programming you should get some great new ideas from it. Again it is a flippy with a whole host of demos plus a documentation file (DV80 format) of 112 sectors.

That's it for disk users this month. For tape users there will just be the one tape this month but it contains 7 good programs. Here is a brief description of each:

TAPE 1987/3 - all Extended Basic.

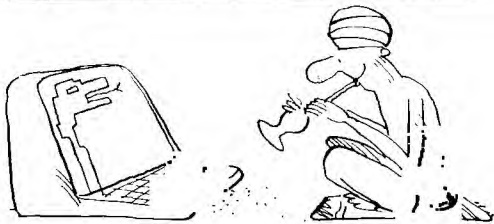
ICEWALL3D - a well programmed game of the Zaxxon variety where you have to destroy the approaching enemy ships. There are some excellent 3D graphics affects in the program.

MISTER T - a funny sort of game with the sole object being to avoid or catch items thrown at you. Well done though and it gets harder to play as you progress through the game.

BALLOON WAR - this game takes a bit of learning before you feel comfortable with it. The basic object is to drift in your balloon dropping bombs on enemy tanks etc. Nice graphics.

BLACKJACK - this one, and the next three are from the same package and are four of the best card simulations I have seen on the TL. The other 3 are CRAPS, POKER and HI/LO. I think you will like these and I'm sure you all know how to play. Thankfully they all play to the rules of the games.

Well that's about it for this month. More news and releases next month



TYPE IT!



```

100 DIM A$(611)
110 CALL CLEAR
120 PRINT " *****
*****";TAB(5);"*****";TAB(24);
**";" * THE FILE MANAGER
**"
130 PRINT TAB(5);"*****";TAB(24)
;"**";" *****
**"; : : :
140 PRINT " PRESS 1 TO READ
FROM TAPE": : " PRESS 2 TO WR
ITE ON FILE": : " PRESS 3 TO
RECORD FILE": :
150 PRINT " PRESS 4 TO SEARC
H RECORDS": : : : " Copy
right 1983 Inter-8"
160 CALL KEY(0,K,S)
170 IF (S=0)+(K<49)+(K>52)TH
EN 160
180 IF (NN>0)*(K=49)THEN 160
190 IF ((NN=0)*(K=52))+((NN=
0)*(K=51))THEN 160
200 CALL CLEAR
210 ON K-48 GOSUB 1510,230,1
340,700
220 GO TO 110
230 REM ADDING TO FILE
240 IF NN>0 THEN 450
250 PRINT "NAME THE VARIOUS
CATEGORIES OF YOUR FILE": :
"(PRESS ENTER WHEN FINISHED)
": : :
260 C=0
270 C=C+1
280 IF C<10 THEN 330
290 PRINT : : " SORRY... NO
MORE ALLOWED.": : :
300 FOR DELAY=1 TO 500
310 NEXT DELAY
320 GO TO 440
330 PRINT : : "CATEGORY";C;":
";
340 INPUT "":NAME$(C)
350 IF LEN(NAME$(C))<21 THEN
390
360 CALL SOUND(160,220,0)
370 PRINT : "TITLE TOO LONG -
TRY AGAIN.": : :
380 GO TO 330
390 IF NAME$(C)="" THEN 440
400 IF NAME$(C)<>" " THEN 27
0
410 PRINT : "THE ABOVE ENTRY
HAS BEEN CANCELLED. ENTIE
R CORRECTED NAME: "
420 C=C-1
430 GO TO 330
440 C=C-1
450 CALL CLEAR
    
```

XB SCREEN COLORS

Reprinted from Tacoma 99ERS
USERS GROUP Newsletter.

Here's a super short, super FAST, assembly routine for BASIC that allows you to change screen and character colors instantaneously! There's lots of possible uses for the thing including games, but the real feature of this program is that it changes the color of the EDIT Mode screen as well! Yep, no more black on cyan if you don't want to! The color change is inserted into the USER-DEFINED INTERRUPT and is constantly "re-performed" every 1/60 of a second. This makes it seem like the EDIT Mode screen has been changed. In order to return control of the color commands CALL SCREEN and CALL COLOR, you must load the User-Defined Interrupt with zeros. Along with the program that loads in the original routine, below are two demo routines to show off your new screen colors. Of course, you don't need a program to change to screen colors once the original file is loaded. All you have to do is poke a single byte value into CPU address 9460. This value is found by doing the following: Foreground color (0 - 15) x 16 plus Background color (0 - 15). For instance, to set the screen to black and the characters to white you'd do the following:
15 x 16 + 1 = 81
CALL LOAD(9460,81)
NOTE:0=transparent,15=white

PROGRAM #1 : SCRNCOLR/X

```

100 ! *****
110 ! * :
120 ! * SCREEN COLOR *
130 ! * :
140 ! *****
150 !
160 !
170 !
180 !
190 !
200 CALL CLEAR :: CALL INIT
210 MEM=9459
220 !
230 ! *LOAD IN PROGRAM*
240 !
250 FOR I=1 TO 50
260 READ X
270 CALL LOAD(MEM+I,X)
280 NEXT I
290 !
300 ! *START UP PROGRAM*
310 !
320 CALL LOAD(8194,37,38,"
,-31804,36,246)
330 END
340 !
350 ! *PROGRAM DATA*
360 !
370 DATA 244,0,2,1,0,135,20
8,96,36,244,216
380 DATA 1,140,2,6,193,216,
1,140,2,2,1,0,72
390 DATA 216,1,140,2,6,193,
216,1,140,2,2
400 DATA 0,0,32,216,32,36,2
44,140,0,6,0
410 DATA 22,251,4,91
    
```

PROGRAM #2: CLRDEMO1/X

```

100 ! *****
110 ! * :
120 ! * COLOR CHANGE *
130 ! * :
140 ! * DEMONSTRATION *
150 ! * :
160 ! *****
170 !
180 !
190 !
    
```

```

200 !
210 !
220 ! *-----*
230 ! NOTE:
240 ! YOU MUST HAVE
250 ! ALREADY LOADED &
260 ! RUN "SCRNCOLR/X"!
270 ! *-----*
280 !
290 CALL CLEAR :: RANDOMIZE
::DIM C$(15)
300 M$="Screen Color Change
Demo"
310 !
320 FOR C=0 TO 15 :: READ C
$(C):: NEXT C
330 !
340 !
350 DISPLAY AT(1,1):M$ :: D
ISPLAY AT(2,1):RPT$(" ",LEN
(M$))
355 DISPLAY AT(10,1):"FRGRM
COLOR:"
356 DISPLAY AT(16,1):"BKGRM
COLOR:"
360 !
370 FC=INT(RND*15)
380 BC=INT(RND*15)
390 DISPLAY AT(10,14):C$(FC)
400 DISPLAY AT(16,14):C$(BC)
410 !
430 CALL LOAD(9460,CVAL)
440 !
450 GOTO 370
460 !
470 DATA Transparent,Black,
Medium Green,Light Green
480 DATA Dark Blue,Light Bl
ue,Dark Red,Cyan
490 DATA Medium Red,Light R
ed,DARK Yellow,Light Yellow
500 DATA Dark Green,Magenta
,Gray,White
    
```

PROGRAM #3:CLRDEMO2/X

```

100 ! *****
110 ! * :
120 ! * COLOR CHANGER *
130 ! * :
140 ! * *****
150 !
160 !
170 !
180 !
190 !
200 ! *-----*
210 ! NOTE:
220 ! YOU MUST HAVE
230 ! ALREADY LOADED &
240 ! RUN "SCRNCOLR/X"!
250 ! *-----*
260 !
270 CALL CLEAR
280 M$="SCREEN COLOR CHANGE
"
290 IS="#" ***** ## ##
*****"
295 BC=5 :: FC=15
300 !
310 DISPLAY AT(1,1):M$ :: D
ISPLAY AT(2,1):RPT$(" ",LEN
(M$))
320 !
330 FOR I=1 TO 8
340 READ C1,M1$,C2,M2$
350 DISPLAY AT(3+(2*I),1):O
SING I$:C1,M1$,C2,M2$
360 NEXT I
370 !
380 DISPLAY AT(22,1):"BKGRN
D:";BC :: ACCEPT AT(22,9) V
ALIDATE(DIGIT)SIZE(-2)BEEP
Happy Computing
RON PREWITT
    
```

This program
continued on page 17

```

100 REM 99 ARTIST
110 REM MARCELLO ZANNINI
120 REM RIDEFINITI
130 REM BASIC OR EXT BASIC
140 CALL CLEAR
150 REM 99 RIMINI VERSION
160 REM DEF. COLOR E SOGGETT
0
170 CALL COLOR(1,7,16)
180 CALL CHAR(32,"")
190 CALL CHAR(33,"0000001818
")
200 CALL CHAR(35,"007E7E6666
7E7E")
210 CALL CHAR(36,"1F13131FFF
FFFF66")
220 CALL CHAR(37,"98987E3D3C
242462")
230 CALL COLOR(2,6,16)
240 CALL CHAR(40,"183C66C3C3
663C18")
250 CALL CHAR(41,"8142241818
244281")
260 CALL CHAR(42,"FFFFFFF7F7
FFFFFF")
270 CALL CHAR(43,"2424FF2424
FF2424")
280 CALL CHAR(44,"2424E70000
E72424")
290 CALL CHAR(45,"E7C3993C3C
99C3E7")
300 FOR I=3 TO 8
310 CALL COLOR(I,2,16)
320 NEXT I
330 CALL CHAR(58,"181818FFFF
181818")
340 CALL CHAR(61,"2424242424
242424")
350 CALL CHAR(63,"0000FF0000
FF")
360 CALL CHAR(64,"2424272020
3F")
370 CALL CHAR(67,"0000001818
")
380 CALL CHAR(69,"2424E40404
FC")
390 CALL CHAR(68,"0000FC0404
E42424")
400 CALL CHAR(83,"00003F2020
272424")
410 CALL CHAR(90,"0000001808
1")
420 CALL COLOR(9,4,16)
430 CALL CHAR(96,"E7C3993C3C
99C3E7")
440 CALL CHAR(97,"0609FF0810
1")
450 CALL CHAR(98,"0000996600
9966")
460 CALL CHAR(99,"6699182424
189966")
470 CALL COLOR(10,15,9)
480 CALL CHAR(104,"FF8181818
18181FFF")
490 CALL CHAR(105,"FF1818181
81818FF")
500 CALL CHAR(106,"118844221
1884422")
510 CALL COLOR(11,14,16)
520 CALL CHAR(112,"FF7F3F1F1
F3F7FFF")
530 CALL CHAR(113,"FFFEFCF8F
8FCFEFF")
540 CALL CHAR(114,"FFFFFFF7F
FFFFFFF")
550 CALL CHAR(115,"FEFFFCFC
FFFEFF")
560 CALL COLOR(12,12,16)
570 CALL CHAR(120,"050519FE3
82CC602")
580 CALL CHAR(121,"007E42427
E5A7E24")
590 CALL CHAR(122,"F8A8A8F8F
FFFFFF66")
600 REM =====
610 REM
620 CALL HCHAR(1,1,35,32)
630 CALL VCHAR(2,32,35,23)

```

```

650 CALL VCHAR(2,1,35,23)
660 CALL HCHAR(24,1,35,31)
670 FOR Y=2 TO 20
680 CALL HCHAR(Y,2,32,30)
690 NEXT Y
700 REM =====
710 REM
720 A$="1Z2$3:4@5S6=7E8D9?0
Q(T,Y*U+I)O-P`AaFbGcHhJiKjLp
;qVrBsMxZyCz"
730 A=LEN(A$)
740 B=2
750 FOR I=1 TO A STEP 2
760 CALL HCHAR(22,B,ASC(SEG$
(A$,I,1)))
770 CALL HCHAR(23,B,ASC(SEG$
(A$,I+1,1)))
780 B=B+1
790 NEXT I
800 REM =====
810 REM
820 Y=16
830 VR=16
840 W=16
850 Z=32
860 X=16
870 GOTO 1060
880 REM =====
890 REM
900 CALL KEY(O,K,S)
910 IF S=0 THEN 900
920 IF K<44 THEN 900
930 IF K>59 THEN 950
940 ON K-43 GOTO 1000,900,10
20,900,1570,1590,1610,1490,1
550,1470,1510,1430,1450,1530
,900,1110
950 IF K<65 THEN 900
960 IF K>79 THEN 980
970 ON K-64 GOTO 1250,1150,1
930,1770,1650,1270,1290,1170
,1370,1190,1210,1090,1040,10
60,1330
980 IF K>90 THEN 900
990 ON K-79 GOTO 1230,1350,1
810,1730,1310,1410,1130,1870
,1690,1390,1990
1000 SUJ=121
1010 GOTO 2070
1020 SUJ=122
1030 GOTO 2070
1040 SUJ=120
1050 GOTO 2070
1060 M=2
1070 N=N+1
1080 IF N=1 THEN 2140 ELSE 9
00
1090 SUJ=112
1100 GOTO 2070
1110 SUJ=113
1120 GOTO 2070
1130 SUJ=114
1140 GOTO 2070
1150 SUJ=115
1160 GOTO 2070
1170 SUJ=104
1180 GOTO 2070
1190 SUJ=105
1200 GOTO 2070
1210 SUJ=106
1220 GOTO 2070
1230 SUJ=96
1240 GOTO 2070
1250 SUJ=97
1260 GOTO 2070
1270 SUJ=98
1280 GOTO 2070
1290 SUJ=99
1300 GOTO 2070
1310 SUJ=44
1320 GOTO 2070
1330 SUJ=45
1340 GOTO 2070
1350 SUJ=40
1360 GOTO 2070
1370 SUJ=41

```

```

1380 GOTO 2070
1390 SUJ=42
1400 GOTO 2070
1410 SUJ=43
1420 GOTO 2070
1430 SUJ=69
1440 GOTO 2070
1450 SUJ=68
1460 GOTO 2070
1470 SUJ=83
1480 GOTO 2070
1490 SUJ=58
1500 GOTO 2070
1510 SUJ=61
1520 GOTO 2070
1530 SUJ=63
1540 GOTO 2070
1550 SUJ=64
1560 GOTO 2070
1570 SUJ=32
1580 GOTO 2070
1590 SUJ=37
1600 GOTO 2070
1610 SUJ=36
1620 GOTO 2070
1630 REM =====
1640 REM
1650 Y=Y-1
1660 IF Y>1 THEN 2120
1670 Y=Y+1
1680 GOTO 900
1690 Y=Y+1
1700 IF Y<21 THEN 2120
1710 Y=Y-1
1720 GOTO 900
1730 X=X-1
1740 IF X>1 THEN 2120
1750 X=X+1
1760 GOTO 900
1770 X=X+1
1780 IF X<32 THEN 2120
1790 X=X-1
1800 GOTO 900
1810 Y=Y-1
1820 IF Y>1 THEN 1830 ELSE 1
670
1830 X=X+1
1840 IF X<32 THEN 2120
1850 X=X-1
1860 GOTO 1670
1870 Y=Y-1
1880 IF Y>1 THEN 1890 ELSE 1
670
1890 X=X-1
1900 IF X>1 THEN 2120
1910 X=X+1
1920 GOTO 1670
1930 Y=Y+1
1940 IF Y<21 THEN 1950 ELSE
1710
1950 X=X+1
1960 IF X<32 THEN 2120
1970 X=X-1
1980 GOTO 1710
1990 Y=Y+1
2000 IF Y<21 THEN 2010 ELSE
1710
2010 X=X-1
2020 IF X>1 THEN 2120
2030 X=X+1
2040 GOTO 1710
2050 REM =====
2060 REM SCRITTURA SOGGETTO
2070 CALL HCHAR(Y,X,SUJ)
2080 M=0
2090 CALL GCHAR(Y,X,B)
2100 N=0
2110 GOTO 2170
2120 IF M=2 THEN 2140
2130 GOTO 2070
2140 CALL GCHAR(Y,X,B)
2150 CALL HCHAR(VR,W,Z)
2160 CALL HCHAR(Y,X,33)
2170 VR=Y
2180 W=X
2190 Z=B
2200 GOTO 900

```



```

10 REM #####
48K JOY DRAW
EXT. BASIC+48 K EXP.
+JOYSTICK
20 REM #####
30 REM BY MARCELLO ZANNINI
40 REM *PLOT* ROUTINE (START
AT MEMORY LOCATION >2700)
50 CALL CLEAR :: DISPLAY AT(
4,3):"PERSONAL JOYSTICK DRAW
ER"
60 DISPLAY AT(10,3):"MACHINE
LANGUAGE UTILITY TO":"";"LO
AD FROM MEMORY EXPANSION ?":
"";"
(Y/N)"
70 ACCEPT AT(15,15)VALIDATE(
"ynYN")BEEP:R$ :: IF R$="N"
OR R$="n" THEN 280
80 DISPLAY AT(18,3):"LOADING
MACHINE UTILITY":"";" F
OR YOUR PERSONAL "":"";"
JOY-DRAWING "
90 CALL INIT :: CALL LOAD(-3
1878,0,"",8196,63,248,"",163
76,80,76,79,84,32,32,39,20):
: MEM=9992
100 FOR I=1 TO 412 :: READ X
:: CALL LOAD(MEM,X):: MEM=M
EM+1 :: NEXT I
110 DATA 0,1,64,65,96,100,12
8,192,255,191,191,0,194,139,
2,0,8,29,4,32,32,40,2,0,8,30
120 DATA 4,32,32,32,2,0,8,31
,4,32,32,32,4,192,2,1,0,3,6,
160,40,42,176,160,39
130 DATA 12,152,2,39,14,26,9
,152,2,39,18,19,6,112,160,39
,9,216,2,39,18,216,2,39,17
140 DATA 4,192,2,1,0,1,6,160
,40,42,152,2,39,15,26,2,112,
160,39,15,208,194,9,51,4150
DATA 192,2,1,0,2,6,160,40,42
,9,50,4,196,209,3,6,196,10,8
4,4,192,208,2,6,192
160 DATA 161,0,4,197,6,195,9
,83,209,67,6,197,4,199,6,194
,9,82,209,194,6,199,5,135,2
170 DATA 6,128,0,6,7,19,2,9,
22,16,252,192,4,4,32,32,40,4
,192,208,1,209,193,6,192180
DATA 10,48,2,1,39,0,2,2,0,8,
4,32,32,44,249,70,39,0,152,7
,39,17,27,16,184
190 DATA 32,39,9,39,18,152,3
2,39,8,39,18,19,34,209,224,3
9,18,4,192,208,160,39,18,2,1
200 DATA 0,3,6,160,40,100,4,
192,208,7,6,192,10,48,2,1,39
,0,2,2,0,8,4,32,32
210 DATA 36,4,192,2,1,0,4,20
8,135,6,160,40,100,192,4,208
,71,4,32,32,32,194,202,4,192
220 DATA 216,0,131,124,4,91,
2,0,30,0,4,32,32,52,4,194,4,
32,32,12,152,32,131,74,39
230 DATA 8,19,6,152,32,131,7
4,39,10,22,3,208,160,131,75,
4,91,152,32,131,74,39,11,22,
234
240 DATA 4,192,208,32,131,75
,6,192,176,160,39,13,6,0,22,
252,176,160,131,76,4,91,2,3,
0
250 DATA 7,216,192,131,74,6,
3,22,252,2,3,0,1,112,160,39,
12,216,32,39,10,131,74,152,2
260 DATA 39,13,26,11,216,32,
39,11,131,74,184,32,39,9,131
,75,112,160,39,13,2,3,0,2,16
270 DATA 242,216,194,131,74,
4,32,32,8,4,91
280 FOR I=96 TO 143 :: CALL
CHAR(I,""):: NEXT I

```

```

290 FOR I=1 TO 14 :: CALL CO
LOR(I,4,2):: NEXT I
300 CALL CLEAR :: PRINT "PLE
ASE RELEASE ALPHA LOCK
...
WAITING FOR YOUR JOY-MOVE" :
: FOR S=1 TO 24 :: PRINT ::
FOR T=1 TO 100 :: NEXT T ::
NEXT S
310 NN=34 :: CALL CLEAR :: C
ALL SCREEN(4):: CALL CHAR(33
,"COCO"):: CALL SPRITE(#1,33
,16,100,100)
320 CALL JOYST(1,XR,YR)
330 CALL MOTION(#1,-YR*2,XR*
2):: CALL MOTION(#1,0,0):: C
ALL POSITION(#1,DOTR,DOTC)::
GOSUB 340 :: GOTO 320
340 IF NN>156 OR DOTR>192 TH
EN RETURN
350 CALL LINK("PLOT",DOTR,DO
TC,NN,KR):: RETURN
100 CALL CLEAR :: CALL SCREE
N(6):: PRINT TAB(6);"SPACE B
OMBS AWAY": : : : : :
: :: PRINT " PRESS ANY KEY
TO CONTINUE":
110 CALL KEY(3,K,S):: IF S=0
THEN 110
120 CALL CLEAR :: CALL SCREE
N(2):: CALL MAGNIFY(3):: RAN
DOMIZE
130 CALL CHAR(128,"0103070B0
A1B2A2B2A2B4A4B4A7B0E0A0080C
0A0A0B0A8A8A8A8A4A4A4BCE0A")
140 CALL CHAR(132,"0000000F1
F357F3F001204020000000000000
0E0F058FCF80050A08")
150 CALL CHAR(136,"0000000000
00001021F7F02070000000000000
0007E84007E840810E")
160 CALL CHAR(96,"0000"&RPT$(
"01",12)&RPT$("0",36),100,R
PT$("0",13)&"1030301"&RPT$("
0",24)&"80C0C08")
170 CALL CHAR(112,RPT$("F",1
6))
180 CALL COLOR(11,6,1):: FOR
X=0 TO 8 :: CALL COLOR(X,15
,1):: NEXT X
190 CALL HCHAR(22,1,112,96):
: SC=0 :: DISPLAY AT(2,3):"S
CORE= 0"
200 CALL SPRITE(#1,128,3,153
,121)
210 CALL JOYST(1,CV,RV):: IF
CV=0 THEN CALL JOYST(2,CV,R
V)
220 CALL MOTION(#1,0,CV*2)::
CALL KEY(1,K,S):: IF S=0 TH
EN CALL KEY(2,K,S):: IF S=0
THEN 230 ELSE GOSUB 380 ELSE
GOSUB 380
230 IF F=0 THEN 260
240 FOR X=4 TO 7 :: IF A(X)=
1 THEN CALL COINC(#X,#2,16,Z
):: IF Z=-1 THEN GOSUB 410 E
LSE CALL COINC(#X,#3,8,Z)::
IF Z=-1 THEN GOSUB 410
250 NEXT X
260 FOR X=1 TO 2 :: IF B(X)=
0 THEN 280 ELSE CALL POSITIO
N(#X+7,R,C):: IF R>165 THEN
CALL DELSPRITE(#X+7):: B(X)=
0

```

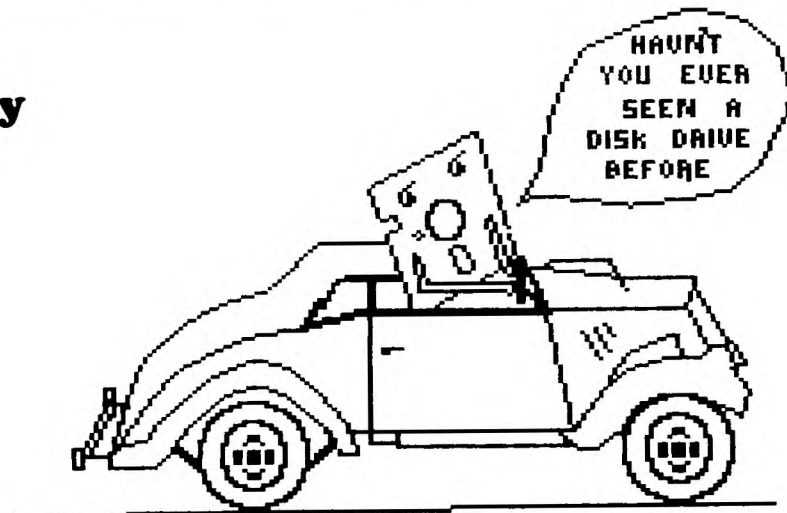
```

270 CALL COINC(#1,#X+7,14,Z)
:: IF Z=-1 THEN 440
280 NEXT X
290 IF A(7)=0 THEN 320 ELSE
IF B(1)=1 AND B(2)=1 THEN 32
0
300 CALL POSITION(#7,R,C)::
FOR X=1 TO 2 :: IF B(X)=0 TH
EN CALL SPRITE(#X+7,100,10,R
,C,16,0):: B(X)=1 :: GOTO 32
0
310 NEXT X
320 Y=Y+1 :: IF Y<>15 THEN 3
50 ELSE Y=0 :: IF A(4)=0 AND
A(5)=0 AND A(6)=0 AND A(7)=
0 THEN 450
330 FOR X=4 TO 7 :: IF A(X)=
0 THEN CALL SPRITE(#X,132,14
,240,RND*256+.5,RND*5+1,RND*
14-7):: A(X)=1
340 NEXT X
350 IF F=0 THEN 210 ELSE CAL
L POSITION(#2,R,C):: IF R>19
2 THEN CALL DELSPRITE(#2)::
F=F-1
360 CALL POSITION(#3,R,C)::
IF R>192 THEN CALL DELSPRITE
(#3):: F=F-1
370 GOTO 210
380 IF F>1 THEN RETURN ELSE
F=F+1 :: CALL SOUND(-200,-1,
5):: CALL POSITION(#1,R,C)::
IF F=2 THEN 390 ELSE CALL S
PRITE(#2,96,16,145,C,-24,0):
: RETURN
390 CALL POSITION(#2,S,S)::
IF S=0 THEN S=2 ELSE S=3
400 CALL SPRITE(#S,96,16,145
,C,-24,0):: RETURN
410 F=F-1 :: CALL POSITION(#
2,S,S):: IF S=0 THEN CALL DE
LSPRITE(#3)ELSE CALL DELSPRI
TE(#2)
420 CALL SOUND(-400,-6,3)::
CALL DELSPRITE(#X):: A(X)=0
:: Y=0 :: IF X=7 THEN SC=SC+
20 ELSE SC=SC+10
430 DISPLAY AT(2,9):SC :: RE
TURN
440 CALL COLOR(#1,16):: CALL
SOUND(-500,-7,0):: SC=SC-50
:: DISPLAY AT(2,9):SC :: CA
LL COLOR(#1,3):: GOTO 290
450 FOR S=1 TO 3 :: FOR X=15
0 TO 130 STEP -2 :: CALL SOU
ND(-99,X,8):: NEXT X :: CALL
SPRITE(#S+3,136,11,249,256,
RND*5+1,-RND*10-5):: NEXT S
460 A(4),A(5),A(6)=1
470 CALL JOYST(1,CV,RV):: IF
CV=0 THEN CALL JOYST(2,CV,R
V)
480 CALL MOTION(#1,0,CV*2)::
CALL KEY(1,K,S):: IF S=0 TH
EN CALL KEY(2,K,S):: IF S=0
THEN 490 ELSE GOSUB 380 ELSE
GOSUB 380
490 IF A(4)=0 AND A(5)=0 AND
A(6)=0 THEN 330
500 IF F=0 THEN 530
510 FOR X=4 TO 6 :: IF A(X)=
1 THEN CALL COINC(#X,#2,16,Z
):: IF Z=-1 THEN GOSUB 410 E
LSE CALL COINC(#X,#3,8,Z)::
IF Z=-1 THEN GOSUB 410
520 NEXT X
530 IF F=0 THEN 470 ELSE CAL
L POSITION(#2,R,C):: IF R>19
2 THEN CALL DELSPRITE(#2)::
F=F-1
540 CALL POSITION(#3,R,C)::
IF R>192 THEN CALL DELSPRITE
(#3):: F=F-1
550 GOTO 470

```

YOUNGER Set with Jenny

The UNDER 18's own page.



**RICHARD
EXTON**

Dear Jenny,

I have designed the above comic picture on the GRAPHX package, hoping that you might use it as the front cover on the next TND or elsewhere.

Thank you

Richard Exton

Dear Jenny,

I hope you like the program I sent you. This is my first letter for 1987. Here are some Adventure hints for Adventure #2: "Pirates Adventure".

YOHO is a very good magic word to use.

If you are stuck in the shack, just remember - Pirates just love Rum!

If you are wondering just where the other exit is to get the lumber out of the cavern, think about the CAVE ridden hills.

To draw the crocodiles away, think about 2 things: then empty bottle; the fish.

I will also give some hints to Return to Pirates Isle.

The alarm clock is not in the ship but under it (trying to feel the underside of the ship might help).

The snail is a water snail.

You need the Pirate to sail the ship.

The rum is where the sign is.

That's it for this month.

All the best,

Vincent Maker

```

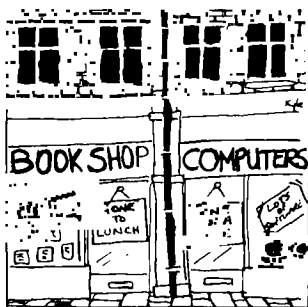
100 F=0
110 J=0
120 RANDOMIZE
130 CALL CLEAR
140 PRINT "GUESS A LETTER OR A NUMBER"
150 PRINT
160 PRINT
170 PRINT " A PROGRAM BY VINCENT MAKER"
180 PRINT
190 PRINT
200 PRINT
210 PRINT
220 PRINT
230 INPUT "WHICH WOULD YOU LIKE?":A$
240 IF A$="LETTER" THEN 260
250 IF A$="NUMBER" THEN 290
260 B=INT(RND*90)+65
270 IF B>90 THEN 260
280 GOTO 300
290 A=INT(10*RND)

```

```

300 IF A$="LETTER" THEN 540
310 CALL CLEAR
320 F=F+1
330 INPUT "HELLO THERE,GUESS A NUMBER BETWEEN 0-10:":D
340 IF D=A THEN 360
350 GOTO 430
360 CALL CLEAR
370 PRINT "CONGRATS! YOU GOT IT IN";F;" GO'S"
380 FOR T=0 TO 500
390 NEXT T
400 INPUT "ANOTHER GO(Y/N)?":FG$
410 IF FG$<>"N" THEN 120
420 STOP
430 IF D>A THEN 450
440 GOTO 500
450 PRINT "TO HIGH!TRY AGAIN!"
460 FOR D=0 TO 300
470 NEXT D
480 GOTO 320
490 IF D<A THEN 500
500 PRINT "TO LOW!TRY AGAIN!"
510 FOR D=0 TO 500
520 NEXT D
530 GOTO 320
540 PRINT "HELLO THERE!YOU MUST HAVE PICKED LETTER"
550 PRINT "PLEASE USE THIS CODE WHEN ENTERING
LETTERS:A=65 B=66 C=67 D=68 E=69"
560 PRINT "F=70 G=71 H=72 I=73 J=74 K=75 L=76 M=77 N=78
O=79 P=80 Q=81 R=82 S=83 T=84 U=85 V=86 W=87 X=88 Y=89
"
570 PRINT "Z=90"
580 J=J+1
590 INPUT "A LETTER-CODE NUMBER,PLEASE:":C
600 IF C=B THEN 650
610 IF C>B THEN 630
620 GOTO 700
630 PRINT "TRY GUESSING CLOSER TO THE START OF THE
ALPHABET..."
640 GOTO 580
650 CALL CLEAR
660 PRINT "CONGRATS,YOU DID IT IN ";J;" GO'S"
670 INPUT "YOU WANT ANOTHER GO(Y/N)?":FG$
680 IF FG$<>"N" THEN 120
690 END
700 IF C<B THEN 710
710 PRINT "TRY GUESSING NEARER THE END OF THE
ALPHABET"
720 GOTO 580

```



T.I. SHOP MARCH 1987

SUPER WIDGET

The following information is from the Victoria (B.C., Canada) 99'er Group's August newsletter.

Over two years ago Johan Vaniaschoot had spoken with me about a large high quality widget type device for more than three modules. The concept was a six or eight slot expansion box with line drivers and a buffered selector for each module. This was to remove the Navarone Widget's worst features, limited expansion and noisy switching.

This idea has hung around and a couple of months ago he and I were talking about it, and I mentioned that I had read about supposed software existing in the TI for multiple pages of cartridges. This built in software would supposedly work with the proper extra hardware. Lending credence to this information was a Millers Graphics newsletter (The Smart Programmer) referring to the address decoding required. Johan and I both had a screen with "REVIEW MODULE LIBRARY" come up from time to time during assembly language development. We talked about the hardware problem and devised a simple scheme to test how the software worked (and if it would work at all).

I loaned Johan a wire wrap and breadboard prototyping system I had built for the cartridge port and with considerable effort Johan managed to locate some 36 pin sockets.

THE SOFTWARE DOES WORK ! This is what appears to be possible :

The TI menu comes up as it usually does and an extra selection is added " REVIEW MODULE LIBRARY ". If this option is chosen the next available cartridge page is displayed on the menu as if it were the only cartridge plugged in and the option " REVIEW MODULE LIBRARY " is also displayed. This action continues in a loop thru all modules until a selection is made of an application. Now this is nice, no need to flick switches and up to 16 modules could be available in a monster box. But there is more.

The GPL (Graphics Programming Language) system is designed so that with this hardware the built in software allows one cartridge to access the devices and calls in another. This allows for example, console basic to access all of the plugged in modules' call routines and device names at one time. " MINIMEM1 ", " MINIMEM2 ", and " SPEECH " and CALL PEEKV, POKEV, LOAD, etc. are all available from basic.

TI FORTH can access " MINIMEM1 " and " MINIMEM2 " and " SPEECH " as devices with no need to switch anything or to modify any console hardware. The operating system in the console handles all accesses transparently.

This was built in from day one with the 99/4 and is on my pre 1983 black and silver console. I don't know for a fact if this is on the newer models, but I suspect it is.

The software during the module library selection finds only those pages that contain rom or grow and rom combined. The slots with rom only (third party stuff) do not come up on the menu. Much like the post 1983 consoles.

The Victoria company Osram Industries is currently developing an inexpensive " Super Widget " to take full advantage of this in console software.

Reprinted from the C.A.U.G ALERT Oct. 1985.

Ed. Comment : Assuming the above comes to pass, imagine the POWER this might add to Extended basic. For example, with the Mini Memory module plugged into the 'super widget' you would have an additional 4K for CALL LINKS to Assembly Language routines as well as CALL PEEKV and CALL POKEV. Now add the TE II module and Extended basic would have easy text to speech. Boggles your mind - doesn't it ?? !!

This is to introduce myself. I am Cyril Bohlsen, the new Merchandise Co-ordinator. For better or worse you will have to put up with me running the the shop for the next year.

SHOP INVENTORY.

At this stage our stock includes:

Micropendiums	\$2.90
(1986-June to December;)(1987 January)	
Spike Protectors	\$29
Console Writer Modules	\$45
Consoles Version 2.2	\$65
1 only Double Disk Drive Cable	\$30
Disk Storage Boxes	\$18.60
Boxes of disks (10)	\$19
Club Software Disks	\$5
Club Software Tapes	\$3
Millers Graphics Smart Prog.Guide.....	\$7.50
32k Matchbox memory expansion kits....	\$45
(Should be available for the March meeting)	

There will be some of the 32k memory exp. boards fully assembled and tested for members at an additional fee of approx \$10 (five solder connections will still have to be made to the console.) Also the grom port you take out of your console will have to be returned to the Club shop.

In the near future we will be carrying some second hand spare parts for the T.I. Console:

- Keyboards
- Grom Ports
- Power Supply Boards
- Ivory Console Cases
- Chips for converting V2 to V1

All the above parts will have been tested by our Technical Co-ordinator.

Being new in this position I would like to hear from my fellow members as to the type of items that they feel the shop should carry.

NOW FOR A FREEBY :

Also in stock are a few copies of the old Magazines

- H.C.M. 1985— Vol 5/2
- H.C.M. 1985— Vol 5/5
- Softex Nov. 83 Vol 1/1
- " Jan. 84 Vol 1/2
- " May. 84 Vol 1/3
- " Sept.84 Vol 1/4
- " Dec. 84 Vol 1/5
- " Mar. 85 Vol 1/6

For the initial release these magazines will be made available to members for the cost of postage only, this is to make it fair for Country members if they require them. The distribution will be first in best served.

METHOD OF CONTACTING SHOP.

- Electric mail.....SHOP
- Written mail.....TISUG P.O.BOX 302, CARLINGFORD
- Phone(02)6395847 7.00 PM to 10.00 PM

METHOD OF PAYMENT

- Bankcard & Mastercard (full details required)
- Cash & Cheque

CYRIL

LINKING EXT'D BASIC-ASSEMBLY



WITH ROSS MUDIE.

HOW TO GET AT TWO PROGRAMS IN THE BASIC GROM
WHEN YOU WANT TO!

by Ross Mudie,

This short article shows how to force the removal of redundant strings in a Basic program when the basic programmer wants it to occur, rather than when it has to happen. This removes the unwanted pause at the worst possible moment in a program.

This article complements a similar item in LINK-IT #5 which appeared in the July 1986 TND for the same feature in extended basic.

The assembly program also shows how to scroll the screen up one line at a time from assembly using program in the Basic GROM.

In the Basic program a method is shown to read the free memory space dynamically as the program is running.

The information of how to gain access to these routines was obtained from TI99/4A INTERN, by Heiner Martin and the BLWP DATA used for the GPLLNK is the actual GROM address (>51A9 on page 196 & >56CD on page 204). The operation of this program is untried on a V2.2 console, but as far as I can ascertain, it should be OK.

```

100 REM SAVE DSK1.TEST
110 CALL CLEAR
120 PRINT "FORCING a GARBAGE COLLECTION"
130 PRINT " by Ross Mudie, January 1987"
140 PRINT :"*BASIC WITH EDITOR ASSEMBLER"
150 PRINT "This Basic program shows how"
160 PRINT "to read memory space free"
170 PRINT "whilst the Basic program is"
180 PRINT "running and fills the memory"
190 PRINT "with redundant strings. You"
200 PRINT "will observe the pause when"
210 PRINT "the Garbage Collection"
220 PRINT "occurs and then you get the"
230 PRINT "opportunity to cause a"
240 PRINT "Garbage Collection to occur."
250 PRINT : "The purpose of this is that"
260 PRINT "using the assembly routine"
270 PRINT "You can control when the"
280 PRINT "Garbage is collected and"
290 PRINT "with proper programming you"
300 PRINT "can eliminate the pauses at"
310 PRINT "the wrong moments.":
320 CALL INIT
330 CALL LOAD("DSK1.BSCSUP")
340 CALL LOAD("DSK1.COMPACT")
350 INPUT "Press enter to continue":E$
360 FOR A=1 TO 45
370 CALL PEEK(-31890,I,M)
380 CALL PEEK(-31974,H,L)
390 PRINT "Free=";STR$( (H*256+L)-(I*256+M)), "Test loop=";
STR$(A)
400 A$="AAAAAAAAAAAAAAAAAAAAAAAAAAAA"
410 A$=A$&A$&A$&A$
420 NEXT A
430 INPUT "PRESS ENTER for compact":E$
440 CALL LINK("COMPACT")
450 CALL PEEK(-31890,I,M)
460 CALL PEEK(-31974,H,L)
470 PRINT "After compact FREE=";STR$( (H*256+L)-(I*256+M)
));:
480 GOTO 360

```

IDT 'COMPACT'
* BASIC Garbage Collection Routine on demand from the
* BASIC program. By Ross Mudie, 2nd January 1987.

DEF COMPAC Source C object COMPACT

REF GPLLNK, VMBW

* Message text with hex 60 added for demo program
TEXT DATA >8A80,>A7AF,>A9AE,>A780,>D4CF * GOING to
DATA >8080,>A7B0,>ACAC,>AEAB,>80C3 GPLLNK c
DATA >CFCD,>DOCI,>C3D4,>808A compact *

SAVRTN BSS 2 Return address storage
LOOPS DATA >0002 Number of loops for SCROLL
LOOPC BSS 2 Scroll counter location

COMPAC MOV R11,@SAVRTN Save return address

LI R0,>02C1 Row 23, Column 2
LI R1,TEXT Execution message
LI R2,28 Number of characters in message
BLWP @VMBW Write message in screen RAM

MOV @LOOPS,@LOOPC Control number of SCROLLs

SCROLL MOVB @LOOPS,@>837C Clear the STATUS byte @>837C
BLWP @GPLLNK
DATA >56CD GROM address of SCROLL

DEC @LOOPC Finished yet?
JNE SCROLL If no, do another

* This is the core part of the code which uses GPLLNK
* to enter GROM 2 at address >51A9 and execute the
* BASIC Garbage Collection routine.

MOVB @LOOPS,@>837C Clear the STATUS byte @>837C
BLWP @GPLLNK
DATA >51A9 GROM address of Garbage Collection

MOV @SAVRTN,R11 Restore return address
RT

END

I hope that this article may spark a few ideas for people who have bought the book INTERN. With GPLLNK if a GROM routine finishes with RTN then it will probably return safely to your calling assembly routine.

I would like to hear from others who find new GPLLNK DATA which access not previously published programs or utilities.



ASSEMBLER

A LOOK AT SPEED by R.A.GREEN

Reprinted from Ottawa T.I.99/4 User's Group Newsletter, January 1986 edition.

The speed of computers is usually specified in "Millions of Instructions Per Second" or simply, MIPS. IBM's newest and fastest computer, the 3090-200, is rated at 30 to 45 MIPS. You would not expect that kind of performance from the TI 99/4 (a \$ 99 bargain), but let's have a look at just how fast our favourite machine is.

There are three areas of CPU RAM in the 99 where machine language programs and their data can reside: (1) the 256 byte scratch pad memory in the console, which we will call "PAD"; (2) the 24K bytes in the memory expansion, which we will call "EXP"; and (3) the 4K bytes in the MINI MEMORY.

The PAD memory is on the 9900's 16 bit data bus and has faster access than the EXP or MM which are both on the 8 bit peripheral bus. I found that the MM and EXP are equivalent.

There are three possible program setups:

- (1) All data and instructions in EXP.
- (2) Data in PAD, instructions in EXP.
- (3) All data and instructions in PAD.

Since the PAD memory is only 256 bytes, option 3 above is quite impractical for any real programs. Most of the data that a program works on is in its 16 workspace registers, so that we could consider the above three setups to be equivalent to the following:

1. Workspace registers and instructions in EXP.
2. Workspace registers in PAD, instructions in EXP.
3. Workspace registers and instructions in PAD.

These are the three situations I experimented with.

The Assembler Language program at the end of this article was the basis of the tests. The sample program is set up for situation (1), with no test instructions. Running this sample would give us the time for the loop overhead which can be subtracted from the later loop timings to give the time for the test instruction itself.

In the sample program, the positions of the Workspace registers and the program instructions is varied by changing the first and second statements. The instruction to be tested is placed inside the TEST loop. The Mini Memory's EASY BUG was used to display test results.

The program uses the VDP interrupt which occurs every 1/60th of a second. The program counts how many times the TEST loop is executed between VDP interrupts. It does this counting 10 times for each test just to be sure the results are repeatable. the whole test for an instruction takes a grand total of 1/6th of a second!

The loop counts obtained are converted to seconds per loop by the formula :

$$\text{seconds / loop} = 1 / (\text{count} * 60)$$

Then, the time to execute the test instruction is obtained by subtraction of the time to execute the loop with no test instruction.

I tested various instructions and found that most of them are the same, the time to execute an instruction being dependent upon the number of words of memory to be fetched and/or stored. I found that the byte instructions were the same speed as the word instructions. That is, ADD BYTE took just as long as ADD WORD. Of course, multiply and divide are the exception to the rule - they take a long time.

Enough said, here are the numbers ! The instruction times are in micro seconds.

Workspace in		PAD	PAD	EXP
Code in		PAD	EXP	PAD
Instruction	Size	(1)	(2)	(3)
DEC REG	1	3.4	4.7	7.5
ADD REG,REG	1	4.7	6.1	10.2
ADD @PAD,REG	2	7.4	8.8	12.8
ADD @EXP,REG	2	8.8	11.5	14.2
ADD @PAD,@PAD	3	10.3	14.2	14.2
ADD @EXP,@EXP	3	14.2	18.2	18.3
MPY REG,REG	2	18.6	18.9	24.2
MPY @PAD,REG	2	20.3	23.0	27.1
MPY @EXP,REG	2	21.6	24.0	28.4

So, how many MIPS is the TI 99/4A? As a good guess, we could average the first six entries in column (2). This gives us an average instruction time of 10 microseconds -- that's 0.1 MIPS. The IBM 3090-200 is only 300 to 450 times faster.

ASSEMBLER SOURCE

```

*
*TITLE :      SPEED TEST PROGRAM
*AUTHORS:    R.A. S.J.GREEN
*
WSP EQU >C000      Workspace Register in EXP
INST EQU >A100     Test instructions in EXP
*
      AORG >A000    Driver program in EXP memory
      DEF START    Define entry point
START LIMB 0       Interrupts off
      LWPI WSP     Load Workspace pointer
*
* Use VDP interrupts for timer
*
      LI R0,INT    R0=user interrupt routine address
      MOV R0,@83C4 Set the interrupt vector
*
* Move test loop to test address
*
      LI R8,INST   R8=test loop address
      LI R9,TEST   R9=Test instructions
      MOV *R9,*R8  Move test loop to test address
      MOV *R9,@2(R8)
      LI R2,DATA   R2 points to 10 counters
      LI R1,10     R1 = 10 times to test
*
* This loop is restarted each VDP interrupt
*
LOOP CLR R0        Clear timing counter register
      LIMB 2       Allow interrupts
      B *R8        Go to the test loop
*
* This is the user interrupt routine
*
INT LWPI WSP      Reload workspace pointer
MOV R0,*R2+      Move count to storage space
DEC R1           Decrement loop counter
JNE LOOP        Add continue for 10 times
BLWP @0         Done, go to title screen
*
DATA BSS 20      10 speed counters
*
* Test loop
*
TEST INC R0      Count times thru loop
      TEST INSTRUCTION
      JMP TEST    Loop till interrupted
      END

```

Reprinted from The Ottawa T.I. 99/4 User's Group
NEWSLETTER September, 1985

A LOOK AT GPLLNK by R. A. Green.

The Operating System of the TI 99/4A consists of code in ROM and in GROM. The code in ROM is assembler language. The code in GROM is TI's proprietary Graphics Programming Language (GPL).
The ROM code has three main functions: interrupt processing, floating point arithmetic and GPL code interpretation. The GROM code has everything else !

There are, in all this GPL code, several very useful routines that can be used by Assembler language programs. The Editor/Assembler and the Mini Memory modules provide a means, called GPLLNK, to access these routines in GROM. The Extended Basic and TI Writer modules do not provide a link to GPL.

I have developed a GPL link routine that will work for all modules. The Assembler source listing is shown below. The code for this routine is a bit tricky, so a few notes for those who want to understand the code may be in order.

1. The workspace registers are already loaded with some necessary values when RAGLNK is called.
2. The first, and only the first, time RAGLNK is called, it searches all GROMs until the hexadecimal value 0FFF is found.
3. The GPL operation code 0F is a call to an assembler language routine. The byte following the 0F, in our case, 0FF gives the table number and entry number in that table. Table number 15 (0F) begins at 8300 in the console CPU RAM, and entry 15 in this table is at address 831E.
4. A GPL CALL stacks the current GROM address then branches to the routine to be called. A GPL RETURN unstacks a GROM address then resumes execution at that address. RAGLNK stacks the GROM address of the 0FFF instruction, then goes to the GPL interpreter to begin execution of the GROM subroutine. When the GPL subroutine does a RETURN, the 0FFF instruction is executed, causing GPL to exit to the assembler language routine whose address is at 831E. This brings 1 back to RAGLNK who returns to his caller.

ASSEMBLER SOURCE LISTING :

```
*TITLE:      GPLLNK Subroutine
*AUTHOR:    R.A.Green
*FUNCTION:   Provides access to the GPL routines,
            no matter which cartridge you are using.
*LINKAGE:   Same as described for GPLLNK in E/A or
            MM manuals, except that the GPL STATUS
            byte need not be reset before calling:
            BLWP @RAGLNK
            DATA GPL-routine-address
*NOTES      This routine depends upon finding the
            value 0FFF somewhere in GROM. This
            value occurs at least 3 times in the
            console GROMs in my machine. 0F is the
            GPL opcode to call an assembler routine.
            *
            DEF RAGLNK
RAGLNK DATA WSP,$+2      Linkage/Transfer Vector
MOV RO,RO                Do we have an address of 0FFF ?
JNE STACK               Jump yes
* Find an occurrence of 0FFF somewhere in GROM
MOV B RO,*R3             Set the GROM address to zero
MOV B RO,*R3
JMP $+4
SRCH1 INC RO             Increment our GROM address
MOV B *R4,R1             Get next GROM byte
SRCH2 CI R1,0FF0         Is it the start of our value ?
JNE SCH1                 Jump no, keep looking
MOV B *R4,R1             Get the byte after 0F
CI R1,0FF0               Do we have 0FFF ?
JEQ STACK               Jump yes, EUREKA !
INC RO                   Bump our GROM address past 0F
JMP SRCH2                And keep looking.
```

```
* Notice that the above loop will not end if we do
* not find an occurrence of 0FFF
* Put our GROM address on the GPL subroutine stack.
STACK INCT *R7           Bump GPL stack ptr at 8373
MOV B *R7,@REG2+1       Get stack ptr into 83xx
MOV RO,*R2               Our GROM address to the stack
MOV *R6,R9               Save contents of 831E
MOV R5,*R6               Put address of BACK into entry
                        >F of table >F.
* Get GPL routine address from CALLER
MOV B R10,@837C         Reset GPL STATUS byte
MOV *R14+,R8             Fetch the GROM address
LWPI >83E0               Switch to the GPL workspace
MOV @REGB,R6            R6=next GROM addr to interpret
B @0060                  Go to GPL interpreter
* Hopefully GPL will come back here
BACK LWPI WSP            Switch back to our workspace
MOV R9,*R6               Restore value in 831E
RTWP                     Return to calling program
* Our workspace registers loaded with interesting stuff.
WSP DATA 0             R0=our special GROM address
DATA 0                   R1 LSB is zero
REG2 DATA >8300         R2=GPL subroutine stack address
DATA >9C02                R3=GROM write address address
DATA >9800                R4=GROM read data address
DATA BACK                R5=address for GPL to come back to
DATA >831E                R6=address of entry >F of table >F
DATA >8373                R7=pointer to GPL subroutine stack
REG8 BSS 2               R8=GROM addr of GPL routine
BSS 2                    R9=saved contents of 831E
DATA 0                   R10=ZERO
BSS 10                   R11-R15
END
```

TND INDEX

Brian Graham

As a service to members I informed all via this Digest last month that I had just completed a complete INDEX to ALL articles in ALL editions of both the SND and the TND from APRIL 1981 to DECEMBER 1986.

The Index is compiled on DBASEIII - an IBM program - for ease of operation and is divided into two parts. Details of its structure was contained in the Jan/ Feb 1987 edition of the TND.

The file is also available as an ASCII text file and I am still seeing if it can be used with a TI Data Base Management System or some other TI program. I will let everyone know how I go.

At the A.G.M. I made available as hard copy through the Club shop some advance copies. They proved quite popular with all who purchased.

The Index is available in four (4) versions as follows :

- (1) 33 page complete index on details of both articles in the SND / TND and the programs from same Cost \$ 5.00
- (2) 33 page complete index on authors of both articles in the SND / TND and the programs from same Cost \$ 5.00
- (3) 7 page complete index on details of all programs in the SND / TND Cost \$ 1.00
- (4) 7 page complete index on authors of all programs in the SND / TND Cost \$ 1.00

These costs may sound high but all profit is going back to the Club and ultimately you the members will be the winners.

A word of warning.... The Index on disk when available will span three (3) disks so if you are waiting for them to be released I would advise you to think seriously about ordering a hard copy to save the time and effort of printing and save the cost of paper.

I believe that there might still be some copies available through the shop. Don't forget postage \$ 1.00 extra.

ORDER YOUR COPY NOW BEFORE IT IS TOO LATE.

```
*****
*
* FORTI MUSIC SYSTEM *
*
*   an ear to a     *
*
* MUSICIANS LANGUAGE *
*
*****
| by ARTO HEINO |
```

When MICHEAL BLACK sent me the CARD to test its potential musicality I was totally totally stunned by the clear professionalism of the FORTI system.

After you slot the card in and plug in your QUAD or STEREO output to a suitable amplifier system, load your FORTI system disk from EDITOR/ASSM and your ready.

The FORTH based language gives you a menu selection on booting, if you haven't read your manual just type ALBUM and it will play the all the tunes on the disk with a GRAPHICAL display of all 12 channels and there pitches and volumes as its playing.

To see the contents of your ALBUM disk type:

```
40 EDIT
```

The 64 column screen with a menu of edit function keys on lower third will be displayed. An excellent editor with copy and move fuctions built in.

There's too much to cover so I'll cover a few examples briefly:

```
SCR #80
0 ( BOOGIE )
1 START
2 VOICE: SNARE 3 OCTAVE
3 2 R: QU A E. A SI A :R
4 E. A SI A E. RE SI A E. RE SI A E. RE SI A FINIS
5
6 <ENV: SNAREENV 14 14 14 10 5 1 0 ENV>
7
8 <PHRASE: BOOGIE-C
9 QU C E G AA BB$ AA G E PHRASE>
10 <PHRASE: BOOGIE-F
11 QU F AA CC DD EE$ DD CC AA PHRASE>
12
13 ->
14
15
```

This is first screen for BOOGIE and this is only the SNARE part for the piece.

Here's a line by line breakdown:

- 0 Remark with title
 - 1 The word START means the actual beginning of the whole program.
 - 2 The word VOICE with a name following (= <31chars) introduces a new voiceline (= <12 channels). The '3 OCTAVE' starts range of the scales used in the tune.
 - 3 'R:' with ':R' at the end is the repeat indicator with the prefix '2' being the number of repeats.
- QU = quarter note
A = A natural
E. = dotted eight note
SI = sixteenth note

4 This line has only A natural note.
RE = rest

FINIS means the end of section.

6 '<ENV:' with 'ENV>' at the end is the envelope (volume*time) for your VOICE.
Each number indicating volume.
(15=loudest)

8 '<PHRASE:' with a name following and 'PHRASE>' at the end is a section of music you can call up any time with the use of its name only.

C E G = Notes
AA = A natural 1 octave up
BB\$ = B flat 1 octave up

You would now write a CONDUCTOR screen to play the phrases in what order and tempo and many other possibilities that FORTI offers.

SOME WORDS USED IN WRITING MUSIC:

ALL	DURATIONS
CONDUCTOR	WH whole note
=REPEAT	HA half note
=MEAS	QU quarter note
+MEAS	EI eight note
=TEMPO	SI sixteenth note
+TEMPO	TH thirty-2nd note
+VOLUME	S4 sixty-4th note
-VOLUME	H. dotted HA
=BASS	Q. dotted QU
=DRUM	E. dotted EI
=TREBLE	S. dotted SI
+FIFTH	T. dotted TH
-FIFTH	3H triple HA
+HALF	3Q triple QU
-HALF	3E triple EI
+OCTAVE	3S triple SI
-OCTAVE	3T triple TH
VOICE	
START	ENV>
FINIS	PHRASE>
<ENV:	:R
<PHRASE	^
R:	FLATS
V	
SHARPS	

NOTES:

A	B	C	D	E	F	G	YOUR KEY
A#	B#	C#	D#	E#	F#	G#	SHARPS
A\$	B\$	C\$	D\$	E\$	F\$	G\$	FLATS
A%	B%	C%	D%	E%	F%	G%	NATURALS
AA	BB	CC	DD	EE	FF	GG	1 OCTAVE UP



MULTICOLOR MODE

by Steven Peacock - WEST JAX 99'ers

In multicolor mode the screen is made up of 48 rows of 64 columns. Each of these blocks are 4 pixels by 4 pixels. One of the best things is that each of these blocks can be any of the 16 colours that the TI computer can make. Also if you want to use sprites, you can. The colour of every two boxes are described by one byte.

The first thing you must do, when using multicolor mode, is number the rows of the screen. In the first four rows the columns are numbered 0 to 31. The next four rows are numbered 32 to 63. This continues for a total of six groups.

ROWS	SCREEN POSITIONS
1- 4	0 1 2.....29 30 31
5- 8	32 33 34.....61 62 63
9-12	64 65 66.....93 94 95
13-16	125 126 127.....125 126 127
17-20	128 129 130.....157 158 159
21-24	160 161 162.....189 190 191

In the multicolor mode, the Pattern Descriptor Table stores the colour of the block, not the character pattern. This table starts at >0800 and ends at >0E00. If you wrote the colour transparent to each of these addresses, the screen would clear.

To put the TI into multicolor mode, write >E8 to VDP. Write only register number one. This is done with the command:

```
LI R0,>01E8
```

Now that the TI computer is in the multicolor mode and you have set up a screen numbering system, you must find the correct byte on the screen. If your starting row value was 12 and the starting column value was 27, here is how you would calculate the correct byte:

```
Y=COLUMN      (Y=27)
X=ROW         (X=12)
A=Y/2        (A=13.5)
B=INT(Y/2)   (B=13)
C=A-B        (C=0.5)
D=X/8        (D=1.5)
E=INT(X/8)   (E=1)
F=D-E        (F=0.5)
G=>0800+E6+B*8+F (G=2408.5)
```

This is the position in the Pattern Descriptor Table. Next you must determine if you need to change the left or right digit. If the value in F is zero, then change the left digit if not, change the right digit.

The command ANDI stands for AND IMMEDIATE. When this command is used to compare two words, the bits that are set in both are set in the new word. For example: If REGISTER 9 has the value of >E318 (1110001100011000) and you ANDI it with >2619 (0010011000011001) the result will be:

```
>E318 1110001100011000
>2619 0010011000011001
Register 9 will now contain:
>2218 0010001000011000
```

The other command A stands for ADD. the command A R8,R9 will add the value in Register 8 and the value in Register 9 and put the answer in Register 9 - the second Register (9). If Register 8 holds >E401 and Register 9 holds >108C the command A R8,R9 results in >E401 plus >108C. The answer is put into Register 9, so Register 9 now holds >f48D. The value in Register 8 is unchanged.

Please note that the A command adds words (registers or a value at an address). The command AI will add a number to the register. For example:

```
A R4,>E09A
A R4,R5
AI R4,31
AI R4,>20
```

The following program starts out by turning the screen black and going into text mode. This is done to set up the screen. Next the screen positions are numbered. Then the Pattern Descriptor Table is cleared to transparent. This loop is also used when you want to clear the screen from the program.

The main loop scans the keyboard for the following keys: W,E,R,S,D,Z,X,C,1,2 and 3. The letter keys are used to draw on the screen. The number keys are used like this: 1 to change screen colour, 2 to change block colour and 3 to clear the screen.

The loop DRAW is the place that computes the location in the Pattern Descriptor Table. If you follow it statement by statement you will find it is the same as I have described above.

One last thing, I have put in a label "RELAY". This is used because the program must jump more than >100 (256 decimal) bytes. All the "RELAY" is used for is a double jump.

```
*****
*PROGRAM BA16A=>BASIC ASSEMBLER
*      MULTICOLOR MODE
*      STEVE PEACOCK
*      via WEST JAX 99'ers
*****
      DEF START
      REF VSBW,VWTR,KSCAN,VSBW
START CLR @>8374      *SETUP KEY SCAN
      LI R0,>0711      *MAKE SCREEN BLACK ON BLACK
      BLWP @VWTR      *BY WRITING >11 TO WRITE ONLY REG 7
      LI R0,>01F0      *PUT IN 40 COLUMN MODE BY
      BLWP @VWTR      *WRITING >FO TO WRITE ONLY REG 1
*****
      START S      . INITIALISATION
      CLR R0      *SCREEN .....ON IS THE START
      LI R7,6      *THERE WILL BE 6 GROUPS OF 128 BYTES
      CLR R5      *REG 5 WILL HOLD VALUE TO BE WRITTEN
LP1  LI R3,4      *THERE ARE 4 LINES IN EACH GROUP
LP2  LI R4,32     *THERE ARE 32 CHARACTERS ON EACH LINE
      MOVW R5,R1   *THERE IS MORE TO BE WRITTEN
LP3  BLWP @VSBW   *WRITE IT
      INC R0      *SCREEN POSITION INCREASED BY ONE
      AI R1,>0100  *ADD ONE TO THE VALUE TO BE WRITTEN
      DEC R4      *DECREASE THE BYTES TO BE WRITTEN TO LINE
      JNE LP3    *IF NOT AT END OF THE LINE STAY IN LOOP LP3
      DEC R3      *DECREASE THE NUMBER OF LINES IN THE GROUP
      JNE LP2    *MORE LINES? STAY IN LOOP LP2
      AI R5,>2000 *START NEXT GROUP 32 (DECIMAL) HIGHER
      DEC R7      *DECREASE NUMBER OF GROUPS
      JNE LP1    *MORE GROUPS? STAY IN LOOP LP1
*****
      CLEAR ALL BOXES TO TRANSPARENT
```



```

CL      LI R0,>0800      *START OF PATTERN DESCRIPTOR TABLE
      CLR R1            *>00 IS TRANSPARENT
LP4     BLWP @VSWB      *WRITE IT
      INC R0            *INCREASE POSITION TO WRITE
      CI R0,>0E00      *>0E00 IS END OF PATTERN DESCRIPTOR TABLE
      JNE LP4          *MORE TO WRITE? STAY IN LOOP LP4
      LI R0,>01E8      *>E8 TO WRITE ONLY REGISTER=MULTICOLOUR MODE
      BLWP @VWTR      *WRITE IT
      SWPB R0          *SWAP BYTE IN REG 0 = >E801
      MOVB R0,@>83D4   *>83D4 = VALUE STORED IN VDP REG 1
      LI R3,32         *COLUMN STARTING POSITION
      LI R4,24         *ROW STARTING POSITION
      LI R5,>0001      *STARTING SCREEN COLOUR (BLACK)
      LI R14,>F000     *STARTING BLOCK COLOUR (LIGHT RED)
*****
*      MAIN LOOP      *
*****
LL      LIM1 2         *FCN QUIT
      LIM1 0          *WILL WORK
      LI R13,>2500     *DELAY LOOP PRINTING BLOCK TO SCREEN
      DEC R13         *INCREASE/DECREASE AS YOU LIKE
      JNE $-2         *COUNT DOWN DONE? NO? AGAIN
      BLWP @KSCAN     *CHECK FOR KEY PRESS
      CLR R1          *REG 1 TO HOLD ASCII VALUE OF KEY PRESSED
      MOV @>8375,R1   *PUT ASCII VALUE INTO RIGHT BYTE OF REG 1
      CI R1,83        *CHECK ASCII VALUE 83 (S KEY)
      JEQ LEFT        *IF EQUAL JUMP TO LEFT
      CI R1,68        *COMPARE IT TO 68 (D KEY)
      JEQ RIGHT       *IF EQUAL JUMP TO RIGHT
      CI R1,69        *COMPARE IT TO 69 (E KEY)
      JEQ UP          *IF EQUAL JUMP TO UP
      CI R1,88        *COMPARE IT TO 88 (X KEY)
      JEQ DOWN        *IF EQUAL JUMP TO DOWN
      CI R1,49        *COMPARE IT TO 49 (1 KEY)
      JEQ SC          *IF EQUAL JUMP TO SC (SCREEN COLOUR)
      CI R1,50        *COMPARE IT TO 50 (2 KEY)
      JEQ BC          *IF EQUAL JUMP TO BC (BLOCK COLOUR)
      CI R1,51        *COMPARE IT TO 51 (3 KEY)
      JEQ CL          *IF EQUAL JUMP TO CL (CLEAR SCREEN)
      CI R1,82        *COMPARE IT TO 82 (R KEY)
      JEQ UPRIT       *IF EQUAL JUMP TO UPRIT
      CI R1,87        *COMPARE IT TO 87 (W KEY)
      JEQ UPLEF       *IF EQUAL JUMP TO UPLEF
      CI R1,90        *COMPARE IT TO 90 (Z KEY)
      JEQ DNLEF       *IF EQUAL JUMP TO DNLEF
      CI R1,67        *COMPARE IT TO 67 (C KEY)
      JEQ DNRIT       *IF EQUAL JUMP TO DNRIT
      JMP LL          *ANY OTHER KEY STAY IN MAIN LOOP LL
*****
LEFT    DEC R3         *MOVE PRINT POSITION 1 TO LEFT
      CI R3,-1        *CHECK TO SEE IF OUT OF BOUNDS
      JNE DRAW        *IF NOT, JUMP TO DRAW
      CLR R3          *IF OUT OF BOUNDS, RESET
      JMP DRAW        *
*****
RIGHT   INC R3         *MOVE PRINT POSITION 1 TO RIGHT
      CI R3,64        *CHECK TO SEE IF OUT OF BOUNDS
      JLT DRAW        *IF NOT, JUMP TO DRAW
      LI R3,63        *IF OUT OF BOUNDS, RESET
      JMP DRAW        *
*****
UP      DEC R4         *MOVE PRINT POSITION 1 UP
      CI R4,-1        *CHECK TO SEE IF OUT OF BOUNDS
      JNE DRAW        *IF NOT, JUMP TO DRAW
      CLR R4          *IF OUT OF BOUNDS, RESET
      JMP DRAW        *
*****
DOWN    INC R4         *MOVE PRINT POSITION 1 DOWN
      CI R4,48        *CHECK TO SEE IF OUT OF BOUNDS
      JLT DRAW        *IF NOT JUMP TO DRAW
      LI R4,47        *IF OUT OF BOUNDS, RESET
      JMP DRAW        *
*****
UPRIT   DEC R4         *MOVE PRINT POSITION 1 UP
      CI R4,-1        *CHECK UP FOR OUT OF BOUNDS
      JNE N1          *IF NOT, CHECK RIGHT
      CLR R4          *IF IS, RESET UP
N1      INC R3         *MOVE PRINT POSITION 1 TO RIGHT
      CI R3,64        *CHECK RIGHT FOR OUT OF BOUNDS
      JLT DRAW        *IF NOT, JUMP TO DRAW
      LI R3,63        *IF IS, RESET RIGHT
      JMP DRAW        *
*****
UPLEF   DEC R4         *MOVE PRINT POSITION UP 1
      CI R4,-1        *CHECK UP FOR OUT OF BOUNDS

```

```

460 PRINT " ENTER DATA:" : :
      :
470 FOR I=1 TO C
480 PRINT :NAME$(I);": ";
490 INPUT "":ITEM$
500 IF ITEM$<>" " THEN 540
510 NN=NN-I+1
520 PRINT : "THE ABOVE RECO
RD HAS BEEN CANCELLED. TYP
E IN CORRECTEDRECORD:"
530 GO TO 470
540 IF (MEM+4*NN>7800)+(NN>5
99)THEN 650
550 NN=NN+1
560 A$(NN)=ITEM$
570 MEM=MEM+LEN(ITEM$)
580 NEXT I
590 PRINT : "-----"
      :
600 CALL KEY(O,K,S)
610 IF S=0 THEN 600
620 IF K=15 THEN 680
630 IF K=13 THEN 470 ELSE 60
0
640 GO TO 470
650 PRINT : " ** MEMORY FULL
**": ; "THIS ITEM CANNOT BE
INSERTEDPRESS ANY KEY TO RET
URN": : :
660 CALL KEY(O,K,S)
670 IF S=0 THEN 660
680 RETURN
690 REM LIST SUBROUTINE
700 CALL CLEAR
710 PRINT "SEARCH UNDER WHIC
H CATEGORY?": : :
720 FOR B=1 TO C
730 PRINT "PRESS";B;"FOR ";N
AME$(B): :
740 NEXT B
750 CALL KEY(O,K,S)
760 IF (S=0)+(K<49)+(K>57)TH
EN 750
770 CALL CLEAR
780 CAT=K-48
790 INPUT "ITEM SEARCHED?
      :
      :SEE$
800 PRINT : " ** SEARCHING **"
810 LIMIT$=SEG$(SEE$,1,LEN(S
EE$))&CHR$(127)
820 II=0
830 SKIP=0
840 TEMP$=CHR$(127)
850 FOR I=CAT TO NN STEP C
860 IF (A$(I)<SEE$)+(A$(I)>L
IMIT$)THEN 910
870 IF TEMP$<A$(I)THEN 910
880 IF (A$(I)=SEE$)*(I<=SKIP
)THEN 910
890 TEMP$=A$(I)
900 II=I
910 NEXT I
920 IF TEMP$<CHR$(127)THEN 9
60
930 CALL HCHAR(23,1,32,20)
940 PRINT " ** END OF FILE **
": :
950 GO TO 1030
960 CALL HCHAR(23,3,32,28)
970 ZH=0
980 FOR PRNT=II-CAT+1 TO II+
C-CAT
990 ZH=ZH+1
1000 PRINT NAME$(ZH)&": ";A$(
PRNT)
1010 NEXT PRNT
1020 PRINT : "-----"
      :
1030 CALL KEY(O,K,S)
1040 IF S=0 THEN 1030
1050 IF K=6 THEN 700
1060 IF K=13 THEN 790

```

```

N2   JNE N2           *IF NOT, CHECK LEFT
      CLR R4           *IF IS, RESET UP
      DEC R3           *MOVE PRINT POSITION 1 TO LEFT
      CI R3,-1        *CHECK LEFT FOR OUT OF BOUNDS
      JNE DRAW        *IF NOT, JUMP TO DRAW
      CLR R3           *IF IS, RESET LEFT
      JMP DRAW        *
*****
DNRIT INC R4           *MOVE PRINT POSITION DOWN 1
      CI R4,48        *CHECK DOWN FOR OUT OF BOUNDS
      JLT N3           *IF NOT, CHECK RIGHT
      LI R4,47        *IF IS, RESET DOWN
N3   INC R3           *MOVE PRINT POSITION 1 TO RIGHT
      CI R3,64        *CHECK RIGHT FOR OUT OF BOUNDS
      JLT DRAW        *IF NOT, JUMP TO DRAW
      LI R3,63        *IF IS, RESET RIGHT
      JMP DRAW        *
*****
DNLEF INC R4           *MOVE PRINT POSITION DOWN 1
      CI R4,48        *CHECK DOWN FOR OUT OF BOUNDS
      JLT N4           *IF NOT, CHECK LEFT
      LI R4,47        *IF IS, RESET LEFT
N4   DEC R3           *MOVE PRINT POSITION 1 TO LEFT
      CI R3,-1        *CHECK LEFT FOR OUT OF BOUNDS
      JNE DRAW        *IF NOT, JUMP TO DRAW
      CLR R3           *IF IS, RESET LEFT
      JMP DRAW        *
*****
RELAY JMP LL           *A RELAY FOR OUT OF RANGE JUMP INSTRUCTIONS
*****SCREEN COLOUR
SC   CI R5,>000F      *COMPARE TO >F (LAST COLOUR)
      JNE $+4         *
      CLR R5           *IF >F THEN RESET TO 0
      INC R5           *IF NOT >F THEN INCREASE BY ONE
      MOV R5,R0        *MOVE REG 5 INTO REG 0
      AI R0,>0700      *ADD >0700 TO REG 0
      BLWP @VWTR       *WRITE THE NEW SCREEN COLOUR
      LI R13,>0500     *DELAY LOOP
      DEC R13          *
      JNE $-2         *
      JMP RELAY        *
*****BLOCK COLOUR
BC   AI R14,>1000     *ADD ONE TO CURRENT BLOCK COLOUR
      LI R13,>0500     *DELAY LOOP
      DEC R13          *
      JNE $-2         *
*****DRAW
DRAW LI R15,2         *PUT 2 INTO REG 15
      MOV R3,R7         *MOVE COLUMN VALUE INTO REG 7
      CLR R6           *CLEAR REG 6
      DIV R15,R6        *DIVIDE REG 6 BY 2. QUOTIENT INTO REG 7
*****REMAINDER IN REG 7
      MOV R4,R9         *MOVE ROW VALUE INTO REG 7
      CLR R8           *CLEAR REG 8
      LI R15,8         *PUT 8 INTO REG 15
      DIV R15,R8        *DIVIDE REG 8 BY 15. QUOTIENT PUT INTO REG 8
*****REMAINDER IN REG 9
      SLA R6,3         *SLA R?,3=MULTIPLY BY 8
      SLA R8,8         *SLA R?,8=MULTIPLY BY 256
      A R6,R8          *ADD REG 6 AND REG 8. ANSWER IN REG 8
      A R8,R9          *ADD REG 8 AND REG 9. ANSWER IN REG 9
      AI R9,>0800      *ADD REG 9 TO >0800. (>0800 IS START OF PATTERN)
*****DESCRIPTOR TABLE
      MOV R9,R0         *MOVE REG 9 INTO REG 0
      CLR R1           *CLEAR REG 1. THE VALUE READ FROM THE PATTERN
*****DESCRIPTOR TABLE WILL BE PUT INTO REG 0
      BLWP @VSBR       *READ THE BYTE
      MOV R1,R10        *STORE THE BYTE IN REG 1, IN REG 10
      CI R7,0          *CHECK REMAINDER OF FIRST DIVISION TO SEE IF
*****LEFT OR RIGHT BYTE IS TO BE CHANGED
      JEQ LD           *IF ZERO THEN JUMP TO LEFT DIGIT
      ANDI R10,>F000    *RIGHT DIGIT TO BE CHANGED SO CLEAR LEFT DIGIT
*****BUT DO NOT CHANGE THE LEFT DIGIT
      SRL R14,4        *SHIFT ALL DIGIT'S FOUR PLACES TO THE RIGHT
      AB R14,R10        *ADD BYTES
      SLA R14,4        *SHIFT FOUR PLACE TO THE LEFT
AL   MOV R10,R1        *MOVE THE NEW COLOUR INTO REG 1 SO IT CAN BE
*****WRITTEN TO THE PATTERN DESCRIPTOR TABLE
      BLWP @VSBW       *WRITE IT
      JMP RELAY        *JUMP TO RELAY (ELSE OUT OF RANGE ERROR)
LD   ANDI R10,>OF00    *CHANGE THE LEFT COLOUR DIGIT, RIGHT UNCHANGED
      AB R14,R10        *WRITE THE NEW COLOUR TO REG 10. DO NOT SHIFT
*****IS IN THE CORRECT PLACE
      JMP AL           *LEFT DIGIT UPDATED, JUMP BACK AND WRITE IT
      END START        *AUTO START

1070 IF K=15 THEN 110
1080 IF K<>10 THEN 1130
1090 PRINT "*** SEARCHING ***"
1100 SEE$=TEMP$
1110 SKIP=II
1120 GO TO 840
1130 IF K<>3 THEN 1310
1140 IF NN=0 THEN 1030
1150 CALL SCREEN(10)
1160 CALL CLEAR
1170 FOR PRNT=II-CAT+1 TO II
+C-CAT
1180 PRINT A$(PRNT)
1190 MEM=MEM-LEN(A$(PRNT))
1200 NEXT PRNT
1210 FOR SHFT=II-CAT+1 TO NN
-C
1220 A$(SHFT)=A$(SHFT+C)
1230 NEXT SHFT
1240 OPEN PRNT=NN-C+1 TO NN
1250 A$(PRNT)=" "
1260 NEXT PRNT
1270 NN=NN-C
1280 CALL SCREEN(4)
1290 PRINT : "THIS RECORD DEL
ETED- PROCEED"
: : :
1300 GO TO 1030
1310 IF K<>4 THEN 1030
1320 PRINT "INSERT NEW RECOR
DS:"
1330 GO TO 470
1340 REM SUBROUTINE TO LOAD
1350 OPEN #1:"CS1",INTERNAL,
OUTPUT, FIXED 192
1360 PRINT #1:NN,C,MEM
1370 PRINT #1:NAME$(1),NAME$(
2),NAME$(3),NAME$(4),NAME$(
5),NAME$(6),NAME$(7),NAME$(8
),NAME$(9)
1380 LINE$=CHR$(127)
1390 FOR CNTR=1 TO NN
1400 SV=SV+1
1410 IF LEN(LINE$&A$(CNTR))<
191-SV THEN 1460
1420 LINE$=STR$(CNTR)&LINE$
1430 PRINT #1:LINE$
1440 SV=0
1450 LINE$=CHR$(127)
1460 LINE$=LINE$&A$(CNTR)&CH
R$(127)
1470 NEXT CNTR
1480 PRINT #1:STR$(CNTR)&LIN
E$
1490 CLOSE #1
1500 RETURN
1510 REM READ SUBROUTINE
1520 OPEN #1:"CS1",INTERNAL,
INPUT , FIXED 192
1530 INPUT #1:NN,C,MEM
1540 INPUT #1:NAME$(1),NAME$(
2),NAME$(3),NAME$(4),NAME$(
5),NAME$(6),NAME$(7),NAME$(8
),NAME$(9)
1550 FOR RD=1 TO NN
1560 INPUT #1:LINE$
1570 PSN1=POS(LINE$,CHR$(127
),1)+1
1580 CNTR=VAL(SEG$(LINE$,1,P
SN1-2))
1590 FOR R=RD TO CNTR-1
1600 PSN2=POS(LINE$,CHR$(127
),PSN1)
1610 A$(R)=SEG$(LINE$,PSN1,P
SN2-PSN1)
1620 PSN1=PSN2+1
1630 NEXT R
1640 RD=R-1
1650 NEXT RD
1660 CLOSE #1
1670 RETURN
1680 CALL SOUND(60,1400,0)
1690 RETURN

```

TI-WRITER Spricht Auch Deutsch!

by Ben Takach



How to use TI-WRITER's language options. No doubt you have noticed the 8 menu options displayed at the power-up stage of TI-WRITER. Some of you may even ventured further down the list than option 2, and found out that the program seems to continue in the usual manner in the English language. There isn't a single word of reference to these options in the confusingly comprehensive TI-WRITER manual.

- | | |
|---|----------------------|
| 1 | TI BASIC |
| 2 | TI-WRITER |
| 3 | TEXTE-TI |
| 4 | TI-TEXT SYSTEM |
| 5 | TRATTAMENTO TESTI |
| 6 | TI-SKRIVARE |
| 7 | TI-TEKSTVERWERKER |
| 8 | PROCESSADOR DE TEXTO |

Fig.1 TI-WRITER's foreign language menu options.

Yet there are quite a lot of foreign language features in this word processing program, which for some obscure reason was not explained by TI. Some of these languages use characters and symbols not present in the English alphabet. These are assigned to certain shifted and FUNCTION keys when the particular language option is selected. Table 1. shows the language options and the redefined characters.

The screen display of true lower case characters will only be printed if the English language option is selected. Small upper case letters are printed if one of the foreign language options were selected. TI's word processor program enables one to write in one of the six foreign languages. Any of the non standard character is typed in by the appropriate shift or function key combination. The completed file may be saved, edited or formatted in the usual manner. Of course it has to be reloaded using the same option, otherwise the screen display would print the conventional characters only.

TI-WRITER LANGUAGE TABLE.

No.	Lang.	Redefined Characters												
		Shift/FCTN.	2	3	4	6	W	R	T	A	F	G	Z	C
2	English	@	#	\$	~	-	[]		{	}	\	~	
3	French		à				•	ø	ù	é	è	ç		
4	German		ß			ß	Ä	Ü	ö	ä	ü	Ö		
5	Portug.					í	•	é	ò	à	è	ù		
6	Swedish	É	Å	Ü	ü	Ä	Å	Ö	ä	á	Ö	é		
7	Dutch	standard characters used												
8	Spanish	R				í	ç	ñ	•			Ñ		
	Standard CH	@	#	\$	~	-	[]		{	}	\	~	

HARD COPY (PRINT-OUT).

The foreign language options would be of limited use without some means to produce hard copies. Since most of the printers were designed for the international markets, these generally have several language options. Dot matrix printers are usually provided with a number of dip-switches for language selection. The popular Seikosha GP-80 and GP-100 series, which were distributed by many outlets under different brand names, enables the selection of English, German or Swedish characters by two dip-switch combinations. Albeit these are well hidden on the main PC-board, and the printer has to be pulled apart to gain access to them (I have duplicated these externally for convenient access).

The codes do correspond to the TI-WRITER special characters and the printer will print the correct letters.

High quality daisywheel printers could present some problems. One have to purchase an ASCII type daisywheel of the particular language. These are available, but it may take some months and several phone calls to the distributor to get it. Silver Reed is a popular type used by many other printer manufacturers as well as Silver Reed. Silver Reed lists ASCII arranged wheels in many languages and several font types. These cost around \$ 30-35. It is important to remember that it must be an ASCII type. The special characters however will not correspond to the TI-keyboard. Further more, some of the standard symbols will also appear at different locations. Files can only be printed out through the formatter, and the text is preceded by a series of transliterate commands.

A German language text printed by a Silver Reed printer will need the following transliterate commands:

- .TL 126:38 (for the sz character)
- .TL 93:126 (for the Umlaut U)
- .TL 125:92 (for the Umlaut u)
- .TL 92:43 (for the Umlaut O)
- .TL 124:61 (for the Umlaut o)
- .TL 91:34 (for the Umlaut A)
- .TL 123:39 (for the Umlaut a)
- .TL 58:27,89 (for :)
- .TL 59:124 (for ;)
- .TL 34:64 (for ")

Text files written in other European languages using special characters may also be printed if these can be produced from symbols and characters present on the type wheel. For example, the Hungarian language uses most of the special characters present on the Swedish wheel, with exception of the letter a with an apostrophe. This can be printed using the transliterate command .TL 92:97,8,39 (FCTN 2 = a + backspace + apostrophe). This will work fine most of the time. At times however the apostrophe is placed on the letter following the a. This is due to a bug in the program. TI-WRITER does not execute transliterate commands with backspaces immediately. First the line is printed without backspaces and apostrophes, then the printer retraces the line and prints the apostrophes just before the carriage return and line feed characters are transmitted. The program stores the character count and calculates the apostrophe locations by it. If there are other transliterates in the line made up from two ASCII codes not involving backspace then these are counted as two characters and thus the following apostrophes get misplaced.

The program is very useful for foreign language correspondence, where an inexpensive type wheel can turn a standard printer into a foreign language machine. The reproduced German language sample shows some of the special characters used by this language.

Zur Sache ZUS Lenksystem Ausfall.

Eine Ferndiagnose ist gewiss schwierig. Es fehlt einem die Möglichkeit Theorien mit Messungen zu prüfen und eventuell bestätigen. Erhaltung der Messresultaten welche vor Wochen durchgeführt waren erreichen einem meistens in ungünstiger Zeit wenn man sich bereits mit anderen Problemen befasst.

Auch die notwendigen Messgeräte werden selten zur Hand sein.

REGIONAL REPORT:

Reports from our
Regional Home-group
leaders...

'See you there 'cause we care'

The Illawarra Regional Group

This group holds regular monthly meetings on the third Monday of each month at 7.30 p.m. at the Keiraville Public School, Gipps road, Keiraville, opposite the Keiraville shopping centre.

We also hold occasional hardware and other special interest group meetings at irregular intervals. We are offering memory expansion and other simple hardware expansions upon request, and are working on software for systems without disks but with memory expansion. We also offer basic service facilities for members.

The meetings normally start with a tutorial session on Extended BASIC, followed by a talk and demonstration of some topic of interest. This leads to some refreshments while members chat about problems and interests. We maintain various libraries for the use of members.

***** LIVERPOOL REGIONAL GROUP *****

The first meeting of the year was held at HANS ZECEVIC'S home with nearly 20 people in attendance.

JOHN PAINE gave a demonstration of the HORIZON RAM-DISK and updated CORCOMP DISK CONTROLLER DSR EPROM.

It looks like TI members will be delighted at the new CALLS and access SPEED these will afford. JOHN also gave a GRAM-KRACKER tutorial.

PETER SCHUBERT demonstrated his new RS232 STAND-ALONE, with BAUD rates of 19200 & 38400 will open new doors to interfacing. His device can also have 32K & DISK CONTROLLER on board!!!

ARTO HEINO also released a new bit of software 'AIDKEY', adding a new function key (FNC 7) with windowing to your SOFTWARE arsenal.

MARCH MEETING - SATURDAY 14/3/87

HELD AT - ROSS HARDY
15 EXCELSIOR ST.
MERRYLANDS

PHONE - (02) 637 6772
TIME - 1 PM

APRIL MEETING - FRIDAY 10/4/87

HELD AT - STEVEN CARR
4 ELIZABETH WAY
AIRDS, CAMPBELLTOWN

PHONE - (046) 281067
TIME - 7.30 PM

CENTRAL COAST REGIONAL REPORT

Central Coast members, and indeed all members are advised that the meeting scheduled for April will be a tutorial workshop on assembly programming. Guests will include Ross Mudie. Venue and Date are yet to be finalised, however further details may be obtained from Russell Welham, telephone 043.924000.

The March Central Coast RG get together will be held on Saturday 14 March. Details again available from Russell.

```

100 CALL CLEAR :: CALL SCREE
N(12)
110 PRINT "COMPRESSOR/REM RE
MOVER": : : "WANT TO 1. CO
MPRESS OR":TAB(10);"2. REMOV
E REMS OR":TAB(10);"3. EDIT
A PROGRAM?": : : :TAB(9);
"CHOICE?"
120 CALL KEY(3,K,S):: IF S=0
THEN 120
130 IF K=49 THEN 160
140 IF K=50 THEN 480
150 IF K=51 THEN 740 ELSE 12
0
160 DISPLAY AT(4,6)ERASE ALL
BEEP:"PROGRAM COMPACTOR": :
: "USES A PROGRAM 'SAVED'
IN MERGE FORMAT": "USE A
DIFFERENT NAME FOR THE'COMPR
ESSED' FILE!" !COMPRESSL
170 !BY J R DEW
180 ! COMPRESSES THE LINE NU
MBERS FROM A MERGE FILE PROG
RAM
190 DEF GLN(X$)=ASC(SEG$(X$,
1,1))*256+ASC(SEG$(X$,2,1))
200 R$=CHR$(201)! LINE NUMBE
R REFERENCE
210 DIM V$(200)
220 INPUT "FILENAME: DSK1.":
F$ :: OPEN #1:"DSK1."&F$,DIS
PLAY ,INPUT ,VARIABLE 163
230 LINPUT #1:L$ :: LN=GLN(L
$)
240 IF LN=65535 THEN 300
250 SS=1
260 P=POS(L$,R$,SS):: IF P=0
THEN 230 ELSE X$=SEG$(L$,P+
1,2)
270 GOSUB 450 :: IP=V :: IF
V=0 THEN NV=NV+1 ELSE 290
280 V$(NV)=X$
290 SS=P+2 :: GOTO 260
300 RESTORE #1 :: PRINT "COM
PRESSED" :: INPUT "FILENAME:
DSK1.":C$ :: OPEN #7:"DSK1.
"&C$,OUTPUT,VARIABLE 163
310 LINPUT #1:A$ :: IF GLN(A
$)=65535 THEN PRINT #7:A$ ::
GOTO 470
320 IF POS(A$,CHR$(147),1)<>
0 THEN PRINT #7:A$ :: GOTO 3
10 ! DATA
330 LINPUT #1:B$ :: P=POS(A$,
,CHR$(132),1):: IF P<>0 THEN
440 ! IF
340 IF POS(A$,CHR$(154),1)<>
0 THEN 440 ELSE IF POS(A$,CH
R$(131),1)<>0 THEN 440 ! REM
OR !
350 IF GLN(B$)=65535 THEN PR
INT #7:A$ :: GOTO 470
360 X$=SEG$(B$,1,2):: GOSUB
450 :: IF V<>0 THEN 440
370 IF POS(B$,CHR$(150),1)<>
0 THEN 440 ! NEXT
380 IF POS(B$,CHR$(161),1)<>
0 THEN 440 ! SUB
390 IF POS(B$,CHR$(154),1)<>
0 THEN 440 ! REM
400 IF POS(B$,CHR$(131),1)<>
0 THEN 440 ! !
410 IF POS(B$,CHR$(147),1)<>
0 THEN PRINT #7:A$ :: PRINT
#7:B$ :: GOTO 310 ! DATA420
IF LEN(A$)+LEN(B$)>=162 THEN
440
430 A$=SEG$(A$,1,LEN(A$)-1)&
CHR$(130)&SEG$(B$,3,LEN(B$)-
2):: GOTO 330
440 PRINT #7:A$ :: A$=B$ ::
IF SEG$(B$,1,1)=CHR$(255)AND
SEG$(B$,2,1)=CHR$(255)THEN
470 ELSE 330
450 V=0 :: FOR X=1 TO NV ::
IF X$=V$(X)THEN V=X :: RETUR
N
460 NEXT X :: RETURN
470 PRINT #7:CHR$(255)&CHR$(
255):: CLOSE #1 :: CLOSE #7
:: PRINT : : "ENTER 'NEW'
THEN MERGE THE": "COMPRESSED
FILE INTO MEMORY.PROGRAM MA
Y THEN SAVED IN NORMAL MAN
NER."
480 DISPLAY AT(1,10)ERASE AL
L:"THE REMOVER"
490 DISPLAY AT(3,7):"BY GARR
ETT MINEO"
500 DISPLAY AT(7,1):"THE REM
OVER WILL REMOVE ALL TYPES O
F REMARKS FROM YOUR PROGRAM
,INCLUDING THOSE USING T
HE TAIL COMMENT (1)."
510 DISPLAY AT(12,1):"NOTE:
THE REMOVER WILL ONLY WORK O
N FILES PREVIOUSLY SAVED
WITH THE MERGE OPTION."
520 DISPLAY AT(18,1):"FILENA
ME:"
530 ACCEPT AT(18,11)SIZE(10)
BEEP:X$
540 DISPLAY AT(22,1):"LINES
EXAMINED >"
550 X$="DSK1."&X$
560 OPEN #1:X$,DISPLAY ,INPU
T ,VARIABLE 163
570 OPEN #2:"DSK1.CLEAN",DIS
PLAY ,OUTPUT,VARIABLE 163
580 Z$=CHR$(255)&CHR$(255)
590 LINPUT #1:X$
600 C=C+1 :: DISPLAY AT(22,1
7):C
610 IF SEG$(X$,1,2)=Z$ THEN
690
620 IF SEG$(X$,3,1)=CHR$(154
)THEN 680
630 IF SEG$(X$,3,1)=CHR$(131
)THEN 680
640 FOR A=4 TO LEN(X$)
650 IF SEG$(X$,A,1)=CHR$(131
)THEN X$=SEG$(X$,1,A-1)&CHR$(
0):: GOTO 670
660 NEXT A
670 PRINT #2:X$

```

