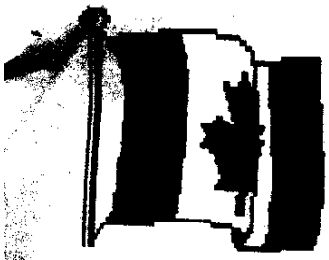
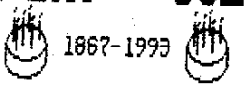


OSHAWA TI 99/4A COMPUTER USERS GROUP

OSHTI

JUNE 1993

CANADA DAY : JULY 1ST!



LAST MONTH'S OSHTI MEETING:

We welcomed back Bernie at the May OSHTI meeting. Thanks go to Doug for hosting the meeting; this gave him an excuse for cleaning up his rec room. We tried to get doug's second hard drive going but it refused to cooperate. After the meeting we did get it formatted. The only problem came shortly thereafter. The program NYARC'S DM5 crashed and we could not get back into the system to transfer files. This only goes to emphasize the point that you must backup everything!

The meeting showed our versatility. We had to switch 2 floppy drives to another PBOX. This doesn't take long but if you need to have double sided drive it must be done. By the way ED,I will be selling a double sided drive probably for the next meeting; CHEAP TOO !

The disk of the month contained a BIORYTHME program. This was a sneaky way to find out everyone's birthday ! It was a lot of fun seeing when our physical, mental and sexual abilities were at their best and worst.

The long awaited FUNNELWEB VSN 5.0 (40 COLUMN) worked for writing and transferring text but it crashed when you tried to go back to the main menu. Tom will try to find out why by the next meeting and have a better working copy.

The updated version of MULTIPLAN VSN 4.03 also proved to be a popular disk for copying.

A tentative date for the oshti annual picnic was set. It is SUNDAY JULY 11TH at TOM'S. Check this date out for the next meeting so that we can firm it up.

The next meeting will be at TOM'S on WED. JUNE 23RD.

TOM

Y.A.P.P. MOUSE LOAD PROBLEM revisited

Steve Andrews (North Bay) and I have found that Run Y.A.P.P. (yet another paint program) will NOT but properly from E/A. The program seems to load OK for some reason the mouse DSR does not.

We have found the way around this.

1- Load YAPP using the LOAD program from XBASIC. This loads the MOUSE DSR properly. However, you cannot use the superspace of the TI unless you are in E/A so we have to exit YAPP.

2-After exiting from YAPP switch over to the E/A cartridge.

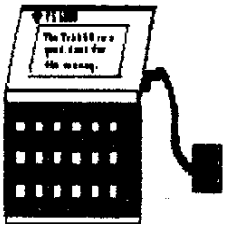
3-Load YAPP under E/A5 OPTION.

When YAPP comes up you will have the proper MOUSE DSR and you can use the SUPERSPACE.

It sounds like a long way round but it seems to be the only way to go.

Maybe CHRIS BOBBITT can fill us in on another way.

JUNE INDEX



TI PS 6600

PC-LINKER

I finally got the PC-LINK ADAPTER for my SMARTZ (64K) ORGANIZER.It cost me $100(CAN) plus tax, but it does everything that I hoped it would do. I can dump the contents of the organizer into an IBM computer through a COM PORT. In fact, this is being typed on it. Although RADIO SHACK no longer sells the PS-6600 called the SMARTZ ORGANIZER, you can still purchase the PS-6600 from large department stores like EATONS in Canada. The PC-LINK enables you to transfer the files to an IBM or CLONE through the COM, PORT (COM1). The PC-LINK comes with excellent software to support the PS-6600. At first, there were rumours that the PC-LINK would not work. In fact, it said in the original manual that you could only pass files to another PS-6600. This is NOT the case.

The software even enables you to write the records to an ASCII file. The ASCII file can then be editted and ported to a TI-99/4A. Of course, the software enables you to edit your files in their original organizer format. The software comes with a 28 page manual written entirely in ENGLISH! (Yes, this is an oddity in Canada.

I don't know why RADIO SHACK discontinued the SMARTZ ORGANIZER just after it was introduced back in OCT.92. My guess is that they were testing the market to see if people would buy it. The recession might also had something to do with it. Now I can type in stuff for the newsletter almost anywhere. I don't have to worry about losing ideas! This is important when you get so many of them.

What I would like to do is try to port this over to the RS232 on the TI. This should not be as big of a problem as one thinks since few wires are involved in sending data. The only thing is what to do with it once you have it.It is interesting that TEXAS INSTRUMENTS still gives its 1-800-TI-CARES number in the manual.

(PS I tried to port the files directly to the TI but of course I didn't know the proper BAUD rate or the PROTOCOL; eg. Imodem etc. No luck here but I can port it easily to the TI as described above.)

Of course, the best way to go is with the software written by TEXAS INSTRUMENTS for the IBM. It is interesting to note that TEXAS INSTRUMENTS has introduced such a product as the TI-6600. I know that Charlie Good of the LIMA GROUP would like one of these to go with his collection.

The TI-6600 is a powerful little machine which has a lot of memory for most users. Sixty-four KILOBYTES is more then enough to meet most user's needs. You can save your entire memory to 3 files in the IBM; TELEPHONE,SCHEDULE and MANUAL.

You can load other files into your organizer. The disks (A 5.25" and A 3. 5") that came with the program have 3 FILES which have HOTEL and TRAVEL INFORMATION for the US and CANADA. This information can be printed out or downloaded into the organizer.

Although you think that you cannot programme on the pc-6600 organizer, you actually can. Just write the programme as a series of lines and end with a c/r. After you transfer the file to the TI, use another programme to read the DV80 file and make it a programme. This was detailed a few months ago in our newsletter.

One thing that you think might limit you in storing your notes(called MEMOS) on the PC-6600 is the 256 BYTE LIMIT PER MEMO. Actually, this is no problem. Just type away. At the end of the memo(signified) by a beep, just press enter and keep typing. The next memo will start automatically.

Another tip when entering memos that you want chain; enter numbers as the first item or better still use A,B,C, as the first byte. This way they will be itemizd one after the other. When you save the content of the memos you can edit the "TEMP.MEM" file before converting to ascii file. Thus you will have the order of the text correct and have only to delete any extra memos.

What I do is to take the ASCII file created and edit it with WORDPERFECT 5.1. This enables me to format it a bit before I transfer it to the TI. The next time I will NOT format it until it gets to the TI. I'm not sure yet which way I will eventually go.

SETTING UP DOUG'S HARDDRIVE (MYAC HFDC)

Since the weather is so nice today, I am typing this article outside under a tree by the pond. Yesterday, Doug and I spent 3 hours working on his harddrive system. We got it up and started to set it up so that it would be menu driven. I thought that the MIAMI BOOT programme would be a good menu system to use but it didn't like to work with the hard drive very well. Maybe it's because of my inexperience with boot but I think it was really designed for use with ram and floppy disk systems.

At any rate we ended up using the recently released menu program which came from Art Gibbens. the programmer who wrote FIRST DRAFT. The menu program enables you to load by chosing options a to z. Although it only has 10 options it can be chained to give as many as you would like. The best part is the fact that it can access the harddrives directly; EG.WDS2.FWEB.FW.

By the way, we also discovered that Funnelweb can be configured for the harddrive under directory FWB. This makes it run very smoothly from the hrd drive.

Doug now has 3 harddrives on his system along with a couple of slimline 5 1/4" drives. Now Doug is the 'POWER USER' in our group.

Now I am getting even more anxious waiting for the new SCSI card to be completed.

MYARC 80 TRACK DISKS:

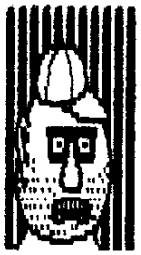
At the LIMA FAIRE I was able to purchase a new chip for my MYARC floppy drive (NOT HFDC just FDC). This chip replaces the FDC chip in the MYARC and allows me to format 80 tracks instead of 40. This means that you can get dsdd with 80 tracks giving 2880 sectors, or double the ordinary DSDD. WOW ! That's 760 K on one diskette. I might add that I had to take a PANASONIC 3 1/2" drive out of my IBM XT to get the extra density. A 360 K drive can't do 80 tracks.

As far as I have been able to assertain, you need A 3 1/2" drive to get 80 tracks. However, since they make 80 track 5 1/4" floppies, there should be 5 1/4" drives that could do the same thing.

One draw-back of course is that you can't share these diskettes with many other pople. A second drawback is that you can't use 40 track formatted diskettes on the 80 track drive. When you set up your disk system, drives 1 to 4 (floppies), MYARC has switches in the controller card to tell it if a drive is 80 tracks. This means that you have to use it as an 80 track drive ONLY! This is a real inconvenience unless you only use the drive occasionally or all the time.

The other problem that I had was in formatting a disk. Funnelweb DISKREVIEW won't work even with QUAD DENSITY. The only disk manager I had left was the MYARC DM III, which I hadn't used in years. This turned out to be wise, as I was to discover. The MDM III will catalog any floppy or ram disk ! That's a real plus right off the bat.

The formatter can be set to 80 tracks and dsdd which gives the 2880 sectors. I guess that I will have to keep using MDM III; it certainy does the job.



INSTALLING A SECURITY SYSTEM:

Thanks to my burglar friends (>:) I have two new steel entry doors and an electronic alarm system. Actually, a special plug for Steve Andrews a TIer in NORTH BAY whose company sells the electronic gyzmos.

As it just so happened, Steve and I went down to LIMA together and he told me that he could get me an alarm system (industrial strength) for a reasonable price. Well he did and now it is properly installed.

After a week delay because of UPS, I got the system and began to read all about it. Well, if you think that doug's harddrive was difficult then you're mistaken; an alarm system is a real test of your skills.

The real problem is that you have to mesh hardware and software and firmware.

The hardware part is the wires that run to your detectors; I have magnetic door devices and a motion detector. The firmware is the EEPROM in the minicomputer and the software is the programming that you do to get it all to work together.

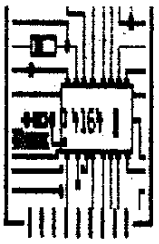
Since I am no expert on the hardware, this was a slow step. Partly because the circuit diagrams have labels like 'NC' and 'NO' which I had no idea about and the use of a load resistor which had to be placed on the last device. It turned out to be very easy but you hate to make a mistake and ruin 50 feet of wire. Of course I had the circuits wrong when I got enough nerve to put the power to it. Then I immediately got an ERROR MESSAGE. It said that the 'FIRE DETECTOR' was improperly attached; I didn't even have one so you can see I was confused. Of course more reading lead to the fact that the 'FACTORY' setting had to be changed. Then I had to 'DISABLE' the ZONES that were not being used each time I set the alarm. Of course it was this time that I realized that my circuits were wrong. A lot of this work was done between 10PM and 1 in the morning so I wasn't thinking to clearly.

Eventually, I got the circuit right and went to use the 'TEST ALARM' program. I thought that this fired the alarm, but I was wrong(at least I think so). Since I wanted to be caught, I did a burglar simulation. SUCCESS ! I was caught moving very nicely. I even got caught trying to escape through the magnetic door catches. The system works very well.

Thanks Steve, this is one sofisticated HOMBRE! I used all of my skills, including careful reading, to get the system to work. it's no wonder that installers are paid so much.

I really didn't think that alarm systems were so complex, I now know that they are, but I know a lot of their secrets too.

Well I guess that I can go in from this lovely weather and upload these files. Thanks to CHARLIE GOOD again for pointing out th value of the ULTRA SMALL COMPUTERS that let you type anywhere!



ASGARD
AMS
SOFTWARE
DEVELOPMENT

AMS DEVELOPMENT SYSTEM INTRODUCTION
=== =========== ====== ===========

Enclosed please find the completed AMS/AEMS Development Kit. This package, which is included with all AMS cards (and soon, AEMS ones as well), is fairware, and may be freely distributed by user groups and other organizations.

This package consists of the following items:

AEMS System "Boot" , version 1.2
AEMS Macro Assembler, version 1.1
AEMS Object Library, version 1.1
AEMS Object Linker, version 1.0

Users of the development system are encouraged to forward a donation to the author, R.A. Green, if they find it useful. Additionally, users are encouraged to write or call with any problems found, so that they can be corrected.

While the amount of labor and time put into this project should impress anyone (our development team, including Art Green, Joe Delekto, Jim Krych, Tony Lewis and myself have been working on this project for over 2 years now), what does this all mean to the TI community? This is a fair question.

Memory management, the design of memory systems, and programming for memory systems is a pretty esoteric discipline. The people involved on this team are the most knowledgeable people in the community on these issues - in fact, I would go so far to say only a 20-25 people in the community can really appreciate their accomplishment, and perhaps only 5-10 could begin to duplicate it.

While the subject is pretty mind-numbing stuff to the average person, if it was solely an exercise to demonstrate technical skills, it would be fair for the community to just congratulate these guys and go back to sleep. However, while the complexity seems unreal, the reason why this was done, and the ultimate results are very important to the future of our community.

Why did we do this? Drastic times call for drastic measures. The TI community is dying because the 99/4A, as-is, can't be pushed much further. More capable software can't be written because there is no memory to put it in. New devices like MIDI interfaces, sound digitizers and multimedia are impossible not because of a lack of speed, but because there is no place to put their data. Graphical user interfaces and desktop publishing, as well as traditional programs like spreadsheets, word processor and databases are hindered more by a lack of memory than anything else. Asgard realized this years ago when some of our more ambitious projects screeched to a halt - because we realized that while we could sketch out what we wanted on paper, there wasn't enough room in the 4A to run it.

While the Geneve was built to address this problem (among several others), we also came to believe that it's a dead- end. Myarc used too many proprietary parts to cost-effectively produce the Geneve. Even if all the system software worked perfectly (which it may never do because of bugs in the hardware), the machine is simply too expensive to make. For this reason, we decided to focus our efforts on the 4A.

We then examined other memory devices available to see if they could be made to work as an Extended Memory Device (EMD). Virtually all of them were totally inadequate for the job. Only OPA's RAMBO came close, and it suffered from a wide range of problems - the major ones being unreliability caused by the fact that the memory card was used for other things (as a RAM-disk), a difficult to use programmers interface, and very slow paging software and hardware. The fact that the designer of the device agreed with our assessment was no comfort.

The only way out of the problem was to go back to the drawing board and start from scratch.

On the hardware end we decided that the only way to make a fast, reliable memory card for programs and data was to make a device dedicated to the purpose. This meant that the card was only usable as an EMD and nothing else. This approach also offered the benefits of dramatically reducing the basic complexity and cost of the hardware design to not much more than that of a 32K card, and of reducing the chance of compatibility problems that has plagued other memory cards. On this basis, we ruled out trying to adapt other cards already available.

After thinking through the hardware, we began exploring what we needed to do on the software end to make it work. We began exploring "segmented memory systems" (as this kind of thing is called) used in the rest of the computer world. After much work, we decided that the approach used by Intel with the 8088 (and TI in their 99/8 and 9900-based minicomputers) was the most applicable - the concept of overlays. Considering how much mileage the PC has gotten out of this idea, with inferior hardware, it seemed like a

good idea to us.

After deciding on an approach, we began designing the hardware and the software. Early on we had to make a decision as to whether or not to put the core controlling software for our device as a DSR on the card or not. After considerable discussion, we decided that a DSR would be a mistake. Why?

1. On a basic level, because the goal of our card was to use memory as memory, not as a device like a RAM-disk or an RS232 card.

2. We found that DSR based software executes much slower than RAM-based software - which is important considering that the core routines would be used constantly. Trying some scheme to load them from a DSR into RAM before executing would increase the hardware complexity and cost, and the chance of a compatibility problem, dramatically.

3. Along the same lines, any DSR would increase the chance for compatibility problems - the last thing we wanted to do was to waste time debugging problems with Myarc or Corcomp products caused by a DSR bug.

4. Lastly, DSRs are fixed. If you find a bug in it, the only way to correct it is to replace it. Consider all the trials and tribulations Myarc and Myarc users went through with periodic EPROM upgrades of the HFDC and the Geneve. Also, if software is written to work around a DSR bug, it may not work with an upgraded DSR.

In contrast, if the software is in RAM, the only thing we would have to do to correct a bug is issue an upgrade. Old programs written for earlier versions of the operating system software would continue to work fine, and new programs could take advantage of new features without worrying about hardware compatibility problems. This is the same reason that Microsoft and Apple load their operating systems from disk, and not from ROM chips.

Finally, on a design level, we felt that we couldn't make the system as easy to program for by trying to shoe-horn everything we needed into a little DSR. The AMS/AEMS development software takes up 5 DS/5D disks - much too much to squeeze into even the largest DSR.

After deciding on approaches for the hardware and software, the only thing left to be done was to implement it. Of course, implementing it DID take 18 months. While the difficulty in designing and building the hardware was nothing to sneeze at, for the most part the real innovation was in the software.

On the software end, the most critical piece was the Linker. This program does all the work of taking pieces of a program and combining them together into a program that can be loaded into, and take advantage of the memory card. The Linker embeds controlling code into each program and its subroutines that makes sure everything is in the right place at the right time. The real innovation is that it is able to do this without the programmer having to explicitly program for it. The only thing the programmer has to do is make sure his program obeys a number of rules that 99/4A programs already generally do anyway. Once done, the Linker will generate a program theoretically up to 16Mb in size.

Another innovation is the Loader, which is built into the System Boot program. The loader takes programs created by the Linker and loads them into the memory card in the pages of memory designated by the Linker. Our particular version also functions as a "memory Manager" - keeping track of memory usage, and allowing several different programs to be loaded at once (as you would a TSR or Device Driver on a PC).

Another critical piece is the Object Library. This file contains a range of subroutines that allow a program to ask for pages of memory and move data around in them. A programmer using these routines in his programs can take advantage of the memory to hold data without explicitly writing code to do so into his/her programs. Further, a Library Manager is included with the package that allows a programmer to customize the library.

Last, but certainly not least, another major component of the package is the Macro Assembler. The most capable assembler available for the 99/4A or the Geneve, this macro assembler further simplifies the programmers life by including common assembly and memory functions in macros. The assembler itself also takes advantage of the memory - giving you much more space for labels (variables), and other such things than you would normally have in an assembly program.

What does all this wonderful technology do that benefits non- programmers? Simply enough, it enables programmers to write programs for the 99/4A that are a match for the capabilities of PC and Mac programs, with no more effort than you would need to do so on a PC. While it doesn't provide an immediate benefit to a non-programmer, in the long run, non-programmers will benefit from what this device lets programmers do - and that they could never do before.

That is why this is so important to the future of the 99/4A. In fact, without a device like this, the 4A probably has no future. With it, the 4A has enough memory to keep programmers occupied and interested for years, and just as importantly, programmers have the tools to use that memory.

Asgard Peripherals is proud of the accomplishment of these talented people, and that we had the opportunity to facilitate and support this kind of effort. We also want to make sure that the product of their labor is not ignored. What we are doing is basic to so many other things we want to do, and to the computer in general. That

is why we are offering to license the design of our basic AMS card to anyone and everyone at practically no cost. That is also why all the development software is available free of charge. Anyone who wants either can phone or call us. Thank you.

Chris Bobbitt
May 12, 1993

Note 1: May be reproduced by anyone if unaltered
Note 2: While AMS/AEMS was not designed for Geneve compatibility, we haven't forgotten 9640 owners. Watch this space for future announcements.

FORMATTER PRACTICE #1

In the APRIL OSHTI newsletter, I discussed the use of the FORMATTER to PRINT your material instead of using PF, PRINT FILE from the EDITOR. The next lesson has some introductory info on some of the commands. The following text is what you would see on the screen of the editor. The OUTPUT form the FORMATTER is seen on page 7 of this newsletter for comparison sake.
These articles are also available at club meetings on disk.

```
.CO PROGRAM #1: FORMATTER
.CO by Tom Jakabfy Apr.93
.LM 5;RM 75;FI;AD
.PA 7
.HE
^^^^^^^^^^^^^^^^&FORMATTER^SERIES^1- T.
JAKABFY
.FO ^^^^^^^^^^^^^^^^^^^^OSHTI  JUN  93
-%-
```

This is the FIRST FILE that you should read about using the FORMATTER.

There are certain things we MUST remember about the formatter.

› a '.' dot as the first character of a line will NOT PRINT. The formatter will then read the characters to the right of the dot and act accordingly.

› the LAST character on the '.' dot line MUST be a c/r or carriage return. Thus, you must type in WORD-WRAP mode (SOLID CURSOR). Make sure that you see the c/r.

› you may have several commands on a line after the dot; but you MUST separate them by a SEMICOLON ';'.

eg '.LM 5;RM 75;FI '

› spaces are important in certain commands. If you were to type in the following command, it will NOT work.

eg '.TL 60 : 10 '

The PROPER way is:

eg '.TL 60:10 '

› CAPITOLS are IMPORTANT. Sometimes the formatter doesn't care, other times it does. ALWAYS use CAPITOLS to avoid problems.

› the character shift 6, 'hat' symbol, prints out as a space. If you want to print a shift 6, then you would have to redefine a character to do this. This is beyond us for the moment, but we can do it by the end. The 'hat' symbol is called a REQUIRED SPACE and is used to join words which must appear together either on the screen or in print. See the next example.

› the symbols shift 2 (at sign) and the shift 7 (ampersand) are PREDEFINED to do @DOUBLE^STRIKE and &UNDERLINE respectively. Placing two @@ symbols or two && symbols together gives only ONE on the print out.

› the shift 8 (star) symbol is also used by the formatter. Placing it in text with a number after it should NOT be done unless you know the effect. One * by itself is OK though.

› printing through the COMMAND LINE gives all the characters you normally see on the screen and the same way they appear... of course the c/r and other CONTROL CHARACTERS still do NOT get printed as symbols.

› type your text in the WORD WRAP MODE... solid cursor. This will ensure that you get a c/r symbol at the end of a line where you want a NEW PARAGRAPH.

FORMATTER SERIES 1- T. JAKABFY

This is the FIRST FILE that you should read about using the FORMATTER.

There are certain things we MUST remember about the formatter.

> a '.' dot as the first character of a line will NOT PRINT. The formatter will then read the characters to the right of the dot and act accordingly.

> the LAST character on the '.' dot line MUST be a c/r or carriage return. Thus, you must type in WORD-WRAP mode (SOLID CURSOR). Make sure that you see the c/r.

> you may have several commands on a line after the dot; but you MUST separate them by a SEMICOLON ';'.

eg '.LM 5;RM 75;FI

> spaces are important in certain commands. If you were to type in the following command, it will NOT work.

eg '.TL 60 : 10

The PROPER way is:

eg '.TL 60:10

> CAPITOLS are IMPORTANT. Sometimes the formatter doesn't care, other times it does. ALWAYS use CAPITOLS to avoid problems.

> the character shift 6, 'hat' symbol, prints out as a space. If you want to print a shift 6, then you would have to redefine a character to do this. This is beyond us for the moment, but we can do it by the end. The 'hat' symbol is called a REQUIRED SPACE and is used to join words which must appear together either on the screen or in print. See the next example.

> the symbols shift 2 (at sign) and the shift 7 (ampersand) are PREDEFINED to do **DOUBLE STRIKE** and <u>UNDERLINE</u> respectively. Placing two @ symbols or two & symbols together gives only ONE on the print out.

> the shift 8 (star) symbol is also used by the formatter. Placing it in text with a number after it should NOT be done unless you know the effect. One * by itself is OK though.

> printing through the COMMAND LINE gives all the characters you normally see on the screen and the same way they appear... of course the c/r and other CONTROL CHARACTERS still do NOT get printed as symbols.

> type your text in the WORD WRAP MODE... solid cursor. This will ensure that you get a c/r symbol at the end of a line where you want a NEW PARAGRAPH.

FAMILY PICNIC SUNDAY : JULY 11

The OSHTI PICNIC we hope will bring out all of the OSHTIers and their spice (spouses) again this year. As those who have attended realize, we have to set a date so that our order for good weather can be filled. We tentatively have chosen Sun. JULY 11th, but this is subject to finallizing at the June meeting.

As usual, the men will be asked to bring a salad, pickle tray, cheese tray etc. Of course you ladies will have to supervise them since they are really computer experts and NOT gourmets. Any ideas about your favourite item can be mentioned at the June meeting.

Who will be the ones to unseat Don and Karen at horseshoes?

Has Bernie mastered the use of the paddle ball since last year ?

Of course, Tom will do his best at not burning the burgers!

Don't forget the swim suits and towels.

See ya there
Tom !

MIND BOGGLER ANSWERS !

The answers to last month's mind bogglers follow. I hope you had a chance to try them first.

2. 7 = W. of the A.W.
2. 7 = Wonders of the Ancient World

3. 1001 = A.N.
3. 1001 = Arabian Nights

4. 12 = S. of the Z.
4. 12 = Signs of the Zodiac

5. 54 = C. in a D. including J.
5. 54 = cards in a deck including jokers

6. 9 = P. in our S.S.
6. 9 = Planets in our Solar System

7. 88 = P.K.
7. 88 = Piano Keys

8. 13 = S. on the A.F.
8. 13 = States on the American Flag

9. 32 = D. at which W.F.
9. 32 = degrees (F) at which water freezes

10. 18 = H. on a G.C.
10. 18 = holes on a golf course

11. 90 = D. in a R.A.
11. 90 = degrees in a right angle

12. 200 = D. for a P.G. in M.
12. 200 = dollars for a pass Go in monopoly

13. 8 = S. on a S.S.
13. 8 = sides on a stop sign

14. 3 = B.M.(S.H.T.R.)
14. 3 = blind mice, see how they run.

15. 4 = Q. in a G.
15. 4 = quarts in a gallon

16. 24 = H. in a D.
16. 24 = hours in a day

17. 1 = W. on a U.
17. 1 = wheel on a unicycle.

18. 5 = D. in a Z.C.
18. 5 = digits in a zip code

19. 57 = H.V.
19. 57 = Heinz varieties

20. 11 = P. on a F.T.
20. 11 = players on a US Football team

21. 1000 = W. that a P. is W.
21. 1000 = words that a picture is worth

22. 29 = D. in F. in a L.Y.
22. 29 = Days in Feb. in a Leap Year

23. 64 = S. on a C.B.
23. 64 = squares on a chess board.

24. 40 = D. and N. of the G.F.
24. 40 = days and nights of the great flood

RETURN TO MINIMEM:

The following article was taken from the BC 99ers' March 99 newsletter. It is a good follow up on my article on MINIMEM. If I don't have room for the entire article here, then I will finish it in the next newsletter.

A Mini Memory Discussion - by Marv Smith

The Mini Memory module was apparently developed to allow users without the expansion box and disk drives to get into assembly programming. It has some unique features such as the line by line assembler. (This is how they avoided the need for memory expansion). Cassette was used in place of disk drives to store programs and data.

A built in battery preserves programs or data in the module even when the power is off. The batteries do wear out but battery holders can be installed to allow battery replacement from time to time without soldering. Two types are available, the first of which was built by our fellow member Werner Ohl and you would have no trouble building one if you chose to do so. The second was discovered by fellow member Ron Warfield which allows the battery to be slipped in.

The module adds: 4K bytes of RAM for program and data storage,

4K of ROM, and;
6k of Graphics read only memory (GROM).

The module contains: additional TI BASIC sub-programs and utility routines that allow you to LOAD and LINK assembly language sub-routines to BASIC programs. The module further contains the means for trouble-shooting your assembly programs by means of EASY BUG.

WHY USE THE ASSEMBLY LANGUAGE? It allows for extraordinary speed and efficiency.

HOW DOES IT DO THIS? It creates machine language code which works directly with the TMS9900 and TMS9918 Video Display Processors. It does NOT have to be interpreted like BASIC.

WHERE DO YOU GET INFORMATION? Unfortunately the Editor Assembly Manual is a technical reference manual and the Mini Memory manual constantly refers you back to the former. Tutorials on how to program in Assembly with Mini Memory are very scarce. One of the few that that deals with Mini Memory and the Line By Line Assembler is "Computes Beginners guide to Assembly Lang. on the TI-994A" by Peter M.L. Lottrup. It is unfortunate the programs were not adequately proof read and this results in time wasted trying to determine what SPACE, PERIOD, OR LINE OF CODE IS MISSING. Mr. Ohl and myself have now completed all the examples and can now provide the corrections to interested parties.

MINOR CONCERNS: One disadvantage of the Line By Line Assembler and Mini Memory is that you do not have a copy of your original source code (note the exception is that part that remains in the buffer) unless you wrote it out first. This makes it difficult to add lines and to make corrections. This disadvantage is offset to some extent by the validation of syntax of each line as it is entered. Dividing your program into sections or blocks will overcome a large part of the disadvantage.

GETTING STARTED. Mini Memory initially has to be loaded from the cassette. The cassette includes:

The Line By Line Assembler
The Old option of the Assembler
The New option of the Assembler
And a demonstration program called LINES.

You load these programs by selecting Easy Bug from the main Menu and typing "L". It loads all of the above. Function and "=" is the quit command.

Once loaded you could type "3" Mini Memory
"2" Run
"LINES" and ENTER and the demonstration program would run. Instead of "LINES" you could key "NEW" to begin a new program or "OLD" to continue a previously started program. However, let us check some things before we start.

SOME TABLE AND PROGRAM LOCATIONS. The default program start location is Hex()17D00. However, if programs are large (such as lines) an earlier start location can be specified. In the case of LINES the start location is >7D06.

The REF/DEF table starts at >7FFF and grows backwards towards >7000. The NEW option of Assembler occupies >7FF8 to >7FFF; and the OLD is in >7FF0 to >7FF7; and LINES has its name in >7FEF to >7FE8. It should be noted that six positions are the program name and the other two positions are the starting address of the program.

We can check this by activating Easy Bug and pressing "M" and the Hex address you wish to see.

If we look at position >7FE8 to >7FED we see Hex 4C 49 4E 45 53 20 which is ASCII 76 73 78 69 83 32 or in English; "LINES ". Note the name is six positions.

Now that we have checked the REF/DEF Table let us run the program. Select (3) Mini Memory
(2) Run
Type in the program name - "LINES"
Press - Enter.

TO BE CONT'D.....

NEXT

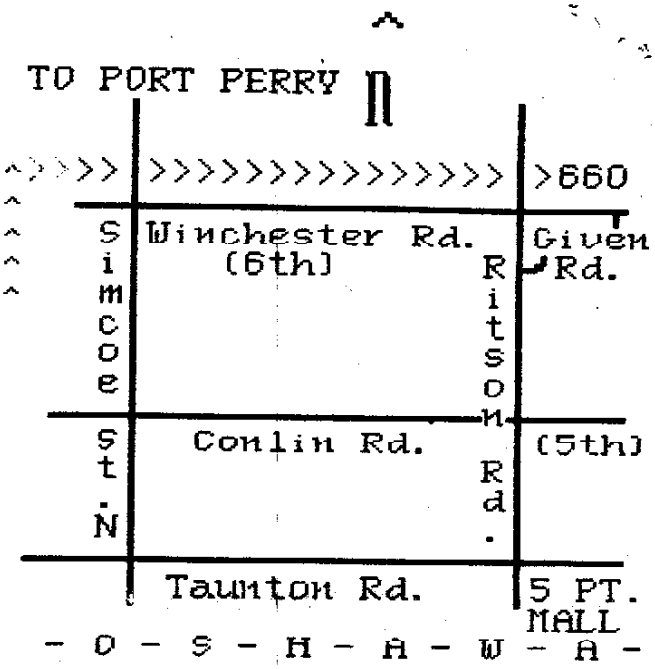
OSHTI meeting

WED. JUNE 23

AT TOM'S
See map >>> >>>>^

PICNIC NEWS
more good stuff...

D.O.M. etc.



OSHAWA TEXAS INSTRUMENTS HOME COMPUTER USERS' GROUP

CHAIR: RAY BLODGETT (579-1767)

TREASURER: KEITH WYARD-SCOTT (723-5758)

LIBRARY/SEC.: DOUG BURLEIGH (579-5109)

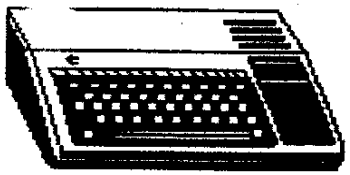
NEWSLETTER EDITOR: TOM JAKABFY (725-7298)

This newsletter was produced on a TI-99/4A using FUNNEL-5.0, PAGE PRO, and PAGE PRO COMPOSER. Occasionally news articles are photocopied.

MEETING TIMES:

The OSHAWA TI USERS' GROUP (OSHTI) meets between the hours of 7:30 and 10:30 pm Location to be named in the newsletter.

OshTI



MEMBERSHIP FEES:

The OSHTI membership is (?) per faeily per year.
Members receive ten(10) newsletters per year. (Jan.-Jun. Sep.-Dec.).
Members also have the use of the club library (CASSETTE + DISK).
VISITORS to club meetings are WELCOME.
Copying charges for disks-of-the-Month are $1(your disk) or $2(our disk)

MAILING ADDRESS:

Tom Jakabfy
660 Given Rd.
OSHAWA, Ont.
L1H 8L7

The OSHTI Users' Group is a Non-profit organization dedicated to encouraging the continued use of the TI/994A for education, entertainment and data management. The club also supports the MYARC 9640 or GENEVE(TI compatible) computer.