Winnipeg Feb 86

WINNIPEG

Newsletter

Febuary's Newsletter

The Winnipeg 99/4 User Group is a non-profit organization formed to meet the needs of Manitoba based Texas Instruments users. The content of this publication does not necessarily represent the view of the Winnipeg 99/4 User Group. This newsletter is one form of communication to keep Manitobans up on Texas Instruments Computers and its clones.

Next General Meeting - Date : March 6th, 1986
Time : 7:00 P.M.
Place: Winnipeg Centennial Library
2nd Floor, Assembly Room

Executive 1986:

President and Educational Co-ordinator:	Jim Bainard	334-5987
Treasurer:	Bill Quinn	837-7758
Newsletter Editor:	Mike Swiridenko	772-8565
Contributing Editor:	Paul Degner	586-6889
Inter-Group Representative and Newsletter Publisher:	Dave Wood	895-7067
Systems Co-Ordinator:	Sheldon Itscovich	633-0835
Public Domain Librarian: 822 Henderson Hwy.	Gordon Richards	668-4804
Module Librarian:	Peter Gould	889-5505
User Programs and Book Librarian:	Mike Swiridenko	772-8565

Mailing Address:

NEWSLETTER EDITER
WINNIPEG 99/4 USERS GROUP
P.O.B. 1715
WINNIPEG, MANITOBA
CANADA, R3C 2Z6

BBS #: (204)-889-1432 SYSOP: Charles Carlson
SETTINGS: 300/1200 baud, 7 bits, 1 stop bit, odd parity
HOURS: 7:00 p.m. to 8:00 a.m. 7 days/week.

EDITORIAL COMMENTS:

Welcome back! Let me start this issue by saying that not everything that reaches me from other groups, and TI sources, can be put into our newsletter. There are many sources of information available for TI owners, that are seen by relatively few members of this club. For instance Paul Degner, Dave Wood and I all receive newsletters from other clubs. Paul receives the MICROPENDIUM and MILLER'S GRAPHICS publications. All of these publications contain much worthwile information. I have seen all sorts of product and software reviews, not to mention programming articles, and tips. Often other TI users voice their problems and solutions for others to see. To have more exposure to what is available in the TI market you should contact one of us and see about borrowing some of the other group's newsletters. The only way you will find out what is going on is if you ask. So ask!

This month we have no feature article. Be it as it may, I hope to have a short series of articles, about writing programs, for the up and coming newsletters. We do, however, have a review by Rick Lumsden, of his SUPER SKETCH Graphics Tablet. Forth users will be interested to know of a General Forth meeting to be held this month. For more details see the READER'S RESPONSE section. Our puzzle and pastime this month is the long awaited word search. That's about all for now! See you next time.

If you have a review, user hints, or helpful programming tips, get them to me for the next newsletter. The deadline for submissions is one week before the date of the group's meeting. Thanks go to all who have submitted items for this issue of our newsletter.

MISCELLANIA:

Miscellaneous news and reminders.

Steve Zabarylo has informed me that he is available to install in-console 32K RAM expansions for the cost of parts plus a modest fee. If interested in this upgrade, published in a past issue of our newsletter, call Steve at 632-9936.

Clubline 99 magazines have arrived and will be available at this meeting for $2 a copy.

Paul will also be selling public domain software packages for those interested.

Our rumored computer program contest will likely be held this year during one of the fall meetings. Categories suggested include: Games, Education, Business, Graphics, Music, and possibly a miscellaneous category. Yet to be decided are rules for judging and entrance, who will judge the entries, and what the prizes will be. Each category could have divisions for the following programming languages: Basic/EXbasic, Assembly Language, Forth, and possibly LOGO.

Thanks go to all who sent in submissions for this month's newsletter, and also many thanks to the clubs which send us newsletters.

Forth followers please read about a General Forth Meeting, in the Reader's Response section.

READER RESPONSE:

To Mike Swiridenko
Newsletter Editor
Winnipeg 99/4 Users Group

While on a recent visit to Grand Forks I had the dubious pleasure of merging files with the two self proclaimed hackers listed below.

Our meeting took place at McGuires a local beer serving establishment that entertains only the higher class members of skid row. Amid the smoke lying over the pool table we were using, and the many Millers and Silver Bullets, I could have swore that I heard them mumble something about free trading of software (will be receiving many discs, including everything available on TI), bulletin boards, discounted hardware sales, and some other things that are now strewn about some bad sectors of my mind.

They look forward to hearing from Winnipeg.

Signed: Doug Hamilton.

Mark A. Gibson
P.O. Box 1574
Grand Forks, N.D.
58206

Jerry Pollard
Ph. 701-775-0114

Also submitted are the following Grand Forks BBS numbers:

NAME	HOURS OF OPERATION	PHONE #
FROZEN BANANA	24 HRS.	701-746-5932
ICE STATION ZULU	24 HRS.	701-594-4924
SNOWBALL	24 HRS.	701-775-8828
NPC NET	4:30pm-8:00am MON-FRI 24 HRS. WKNDS & HOLIDAYS.	701-795-4212

[I've tried the FROZEN BANANA BBS. Mentioned, in the messages I read, were the SNOWBALL, and NPC NET BBSs. Some interesting reading south of the 49'th! Check it out, if you dare. ed.]

WINNIPEG FORTH INTEREST GROUP MEETING

A GENERAL FORTH meeting was announced, on the ARIEL I BBS, near the end of January. The meeting is to be held TUESDAY FEBUARY 11, 1986. Time and place undecided as of the date I typed this.

There were Forth Interest meetings held late last year, and I had opportunity, on a couple of occasions, to attend. The meetings I attended were introductory in nature and aimed at the uninitiated. The structure of the

HELPFUL HINTS AND TIPS!
(FOR THE USERS, BY THE USERS!)

This column features tips brought to my attention from members of this group, other user group's newsletters, and various other sources. **WARNING:** These hints and tips are to be used at your own risk!

ASSEMBLY LANGUAGE:
The following is reprinted from the TINS January newsletter.
ALPHA LOCK by R.A.Green, Ottawa U.G. Newsletter
Have you ever started a game then have to restart because you forgot to release the Alpha Lock? Well, here is a little trick that Assembler Language programmers can use to warn the player, and then wait till s/he releases the Alpha Lock. It will make a programme a little more user friendly.

The following source programme demonstrates the technique - it literally looks at the Alpha Lock.

Assembler Source

```
*TITLE: ALPHA LOCK DETECTION
*AUTHOR: R.A. GREEN
DEF  ALPHA   Define entry pooint
REF  VMBW    VDP Multiple byte write
*
ALPHA    LI   R0,>0062  "ALPHA LOCK TEST"
LI   R1,MSG1
LI   R2,15
BLWP @VMBW
TEST     CLR  R12        CRU Base address of zero
SBZ  21
NOP
TB   7          Test for Alpha Lock
JEQ  NOTON      Jump if Alpha Lock is off
LI   R1,MSGON   "ALPHA LOCK IS ON"
MSG      LI   R0,>00A2
LI   R2,17
BLWP @VMBW
LIMI 2          Let interrupts in
LIMI 0          So QUIT key will work
JMP  TEST       Loop until Alpha lock is released
NOTON    LI   R1,MSGOFF "ALPH LOCK IS OFF"
JMP  MSG        Loop back to display MSGOFF and test again
MSG1     TEXT 'ALPHA LOCK TEST'
MSGON    TEXT 'ALPHA LOCK  IS ON'
MSGOFF   TEXT 'ALPHA LOCK IS OFF'
END
```

Corrections and comments by Frank Geitzler.

Reprinted with corrections from last month's issue. [TINS DEC] It has been found that if the Alpha Lock is up and the function key is pressed, this program indicates that the Alpha Lock is down.

TI-WRITER
Basic programs may be included in letters and articles by merely listing the program to a disk file as follows, LIST "DSK1.PROGRAM", then editting the file using the TI-Writer.

EXTENDED BASIC
You may highlight input areas of a screen by using the following trick:

```
100 CALL CHAR(143,"FFFFFFFFFFFFFFFF") !DEFINES A BLOCK CHARACTER.
110 CALL COLOR(14,7,7) ! GIVES THE BLOCK CHARACTER THE COLOR DARK RED. FOREGROUND AND BACKGROUND.
115 BL$=CHR$(143)      ! A SINGLE BLOCK CHARACTTER.
120 A$=RPT$(BL$,10)    ! BUILDS A STRING OF TEN BLOCK CHARACTERS.
130 DISPLAY AT(24,1):"ENTER SIML DATA: ";A$    ! ASK FOR DATA.
140 ACCEPT AT(24,18)SIZE(-10)BEEP:B$   ! GET DATA FROM SCREEN.
150 BPOS=POS(B$,BL$,1) ! FIND POSITION OF FIRST BLOCK IN THE DATA STRING.
155 ! REMOVE THE BLOCK CHARACTERS FROM THE DATA STRING.
160 IF BPOS<>0 THEN B$=SEG$(B$,1,BPOS-1)
165 PRINT LEN(B$);B$;BPOS  ! THIS DISPLAYS THE DATA STRING AFTER THE REMOVAL OF THE BLOCK CHARACTERS.
```

The above code defines a solid character, colors it red, and uses it in a display message to highlight an area used for data input. The string variable B$ will accept the data value entered. Since the colored blocks are accepted as part of the data they must be removed before the data can be used. This just one example of possible methods that can be used to highlight a screen input area.

meetings consisted of a short tutorial, given by Bruce Walzer an avid Forth entusiast, followed by hands on experience. Several CP/M systems were set up to run FIG-Forth. These systems were used by participants to try examples, and problems. Problems were taken from the book called 'Starting Forth', written by Leo Brodie.
The Forth meetings this year promise to be more diversified as I have heard talk that the coordinators are looking for interesting problems, and techniques to try. If you are at all interested in programming, and the Forth Language in particular, I urge you to attend the meetings if you can. The facilities for learning, when I attended, ~~were excellent, the tutorials were presented professionly, and best of all the meetings were FREE (that may have~~ changed by now, I don't know.). Meetings will most likely begin at about 7:00pm Tuesday nights. Anyone interested can call Les Johnson at 475-0115, for more details.

REVIEWS:

This column presents reviews of materials that may be of interest to the user. The views expressed are the opinions of the reviewers, exclusively.

HARDWARE:

SUPER SKETCH A review by Rick Lumsden

I recently received a SUPER SKETCH tablet this past Christmas and would like to pass on a short review to any of the membership who are considering purchasing this unit. The opinions of this unit are my own and for that reason some of the things I consider detrimental, or a plus, to its operation may be the exact opposite in your own case.

The unit itself operates flawlessly and does everything it is supposed to. You can use any of the sixteen colors at any time and draw with pixel accuracy. There is also a texture feature which will fill an area with a texture in a color of your choice.

There is a choice on the main menu called 'BRUSH' which allows you to chose a certain brush stroke. There are eight different styles to pick from. The 'FILL' command operates as well, and even better, than most I have seen. The 'EXPERT' command moves you to another menu and there you can chose from different drawing modes such as 'RAYS', 'BOX', and so on. This is my first complaint about the unit. The absence of a circle command. Perhaps it is not thought to be necessary, but for myself it would be a welcome addition.

All the options are chosen by moving the cursor, with the wand attached to the unit, and then pushing the appropriate button on the top of the SUPER SKETCH. Indeed very simple to operate. One can use it with ease in a matter of minutes without continually having to look back at the manual for references on a particular keystroke.

SUPER SKETCH is designed to be used with a basic console, with no additional peripherals necessary. The only item you may need is a cassette recorder to save your pictures. Here lies my second complaint. You are unable to save your pictures to disk. This, I believe, was a design feature so that the user that does not have an expanded system could benefit from its use. I would, however, like to have seen an option to use either disk or cassette at your own choosing. I have nto as yet figured a way to dump any of the drawings to the printer but according to informed sources this does not present too much of a problem. There is a module available called SKETCH MASTER which fills the two gaps I have just mentioned, allowing you to save to disk or printer; however, this module requires the use of another peripheral from Naverone, called the Widget.

In conclusion, this is a very good unit for the unexpanded TI. There are better drawing programs around but if you have tried drawing with a joystick you can soon learn to appreciate the tracing features of SUPER SKETCH. The unit is very reasonably priced at $47.95 American through Tex-Comp, or if you or a friend is a member of the National Assistance Group you may purchase it there for $39.95 American.

I hope this will give those of you in the club a better idea of the operation of the unit, and if before buying you would like to see it on operation give me a call and I will bring it out to the next meeting.

[Rick is a frequent contributor to this newsletter and can be reached at his home number, 253-0794. ed.]

PROGRAMMERS CHALLENGE! SEE IF YOU CAN UNDERSTAND AND IMPROVE THE FOLLOWING PROGRAM!

```
100 DIM TONE(100)
105 DEF F1(X)=INT(AN(X)*MS^TONE(K))
110 RESTORE 130
120 FOR I=1 TO 9 :: READ X$ :: CHOICE$(I)=X$ :: NEXT I
130 DATA (1) DURATION,(2) FREQ1,(3) VOL1,(4) FREQ2,(5) VOL2,(6) FREQ3,(7) VOL3,(
8) NOISE,(9) VOL4
140 MS=2^(1/12)
150 CALL CLEAR :: CALL SCREEN(2):: FOR I=0 TO 14 :: CALL COLOR(I,16,2):: NEXT I
160 RESTORE 180
170 FOR I=1 TO 9 :: READ AN(I):: DISPLAY AT(I+10,3):CHOICE$(I);TAB(19);AN(I):: N
EXT I
180 DATA -100,110,0,1100,0,1210,0,-1,0
190 FOR I=58 TO 90 :: TONE(I)=I-58 :: NEXT I :: TONE(44)=28 :: TONE(46)=29
200 CALL KEY(3,K,S):: IF S=0 THEN 200
210 IF K>47 AND K<58 THEN GOSUB 500
220 CALL SOUND(AN(1),F1(2),AN(3),F1(4),AN(5),F1(6),AN(7),AN(8),AN(9))
230 GOTO 200
500 REM GET NEW DATA
510 DISPLAY AT(21,3):CHOICE$(K-48):: ACCEPT AT(21,19)VALIDATE(NUMERIC):M
520 AN(K-48)=M :: DISPLAY AT(10+K-48,19):M
530 RETURN
```

SUBMITTED BY DOUG HAMILTON.

HOW TO KILL AN ORGANIZATION

from: President's Corner, Summit 99'er Users Group

Nora Sorkin, president of the Summit 99'er Users Group in Cuyahoga Falls, Ohio spotted this set of rules on the "FIRECOMM" BBS while he was doing some modeming one evening. With thanks to Nora and FIRECOMM, here they are:

1. Don't attend meetings; but if you do, arrive late.

2. Be sure to leave before the meeting is over.

3. Never offer your opinion at a meeting; wait till you get outside.

4. When at meetings, vote to do everything then go home and do nothing.

5. The next day find fault with your officers and fellow members.

6. Take no part in your organization's affairs.

7. Sit in the back and start up your own meeting with one or more members during discussion periods; if you keep it down low, no one will notice.

8. Get all the organization can give and give nothing in return.

9. Talk cooperation but never cooperate.

10. Never ask anyone to join the organization.

11. Threaten to resign at every opportunity; especially when things are not going your way.

12. If asked to help, always promise to do so but be busy when called upon.

13. Never read anything pertaining to the organization in case you learn something on your own.

14. Never accept an office; better to criticize than be criticized.

15. If in a moment of weakness you find you have gotten yourself on a committee; apply all of the above rules and let the chairman do all of the work.

16. Don't do anything more than you have to and when others give freely and willingly of their time and talents to help the cause, be the first to leap to your feet to remind everyone:

What's wrong with this group is that it's being run by a clique!

The ABOVE WAS TAKEN FROM THE EDMONTON 99'ers APRIL 1985 NEWSLETTER. IT SPEAKS WELL FOR THE REASONS WHY HOBBY AND SPECIAL INTEREST GROUPS BREAK DOWN. IT IS ALSO A GENTLE REMINDER FOR THOSE OF YOU WHO HAVE NOT PAID YOUR MEMBERSHIPS AND STILL WISH TO BELONG TO THE WPG. TI-99/4 USERS GROUP TO DO SO SOON!

Only your participation will make us worth your while. RIGHT?

Quid Pro Quo

by Paul Degner

So this is the beginning of our second month of the new year and so far still no new news on 'Noah' but we still have two more months to go before the first quarter runs out! Oh well, patience breeds eventuality! A reminder to those of you who haven't yet signed on to Charles Carlson's BBS, please do so! The long awaited download section is working smoothly featuring Xmodem transfers. If you need a terminal emulator program please contact Charles or myself. We usually carry the latest updates of Fasterm or the other freeware terminal emulators. Now on to this month's gab.

Briefs:

A few days ago, having some time to kill, I poured over my collection of letters from old TI associates. One letter was composed of photocopies of a few pages from IUG's magazine 'Enthusiast 99' which a dear old friend, Bob McKenzie, sent while I was into my cracking cartridges stage and he thought this would help me. Bill Gronos use to write a column about Assembly language programming for the TI and this is the first of two parts, the latter part will appear in the next issue, of some passages taken from his column that may be of some interest to you.

ASSEMBLY LANGUAGE

Those of you who have played around with Assembly programming quickly learned just how many functions that the BASIC interpreter was happy to take care of for you, with nary a whimper. BASIC acted as your foreman. All you needed to do was give it a set of blueprints and it would instruct the various crews such as floating point conversions, video display handling and peripheral access control to do what you wanted them to do. And all this crew supervision was invisible to you. But foremen have to be payed and BASIC is no exception. The currency of the payment is milliseconds: the bigger the job, the more you are going to have to fork over.

If a contractor wanted to save money, he could lay off his foremen and take over the direct supervision of the workers himself. Of course, he must have the knowledge required to do this, or the project will end up a shambles. This, in effect, is what you do when you program in Assembly language: you fire the foreman and pocket his paycheck. However, many of you quickly realized that the foreman really was earning his exorbitant pay. Taking care of all that supervision is hard work because your crew of workers is a bunch of idiots who have to be told exactly what to do, down to the very last detail.

INTERRUPTS

If your childhood was anything like mine, you were probably at one time or another chided with the phrase: "don't you know it's not polite to interrupt!!" While interrupting a conversation may be rude, computer interrupts are a powerful and vitally necessary tool.

Basically, interrupts work like this: the computer processor is busy working on a job when all of a sudden it is called away to perform some other job. It makes a note of where it is on the first job, sort of using a bookmark, and then begins work on the new job. When the calling job is completed, it then returns to the old job and continues where it left off.

The call to break off what is presently being done in order to tackle a new job is called an interrupt request (IR). IRs are metaphorical flares that the computer processor is constantly searching the sky for. These sky-borne flares do not mark where the new job to be done is at; they all are shot off at the same place. However, the processor knows where to get the job location address from, because that has been agreed on in advance.

Let's make this metaphor just a tad more concrete by comparing the computer processor to a pizza deliverer. At the start of his shift, the pizza person is given a stack of pizzas to deliver. He begins to deliver the pizzas when all of a sudden he spots a flare in the air. He has been well trained and he knows that when the flare goes up he is to drive to a predetermined spot to get futher information. He puts a check mark on his delivery list so he knows where to take the next pizza and heads off to the predetermined location, which happens to be the park bench nearest the drinking fountain at the local picnic ground. At the bench will be a brown paper bag that contains two cards: one card tells him where he is to pick up the instructions for the new job and the other card tells him what to do with the car load of pizzas. An important point to note is that this predetermined location did not contain the actual instructions for the new job, but rather where to find the instructions. The location card says, "Corner of 27th street and Eagle Lane". The "what to do with the pizzas card" says, "Park the truck and take the '56 Cadillac".

The pizza man exchanges the truck for the Caddie and drives over to the indicated address. At that address, he finds the owner's daughter waiting for him and she has his new instructions: "Dad said that you would drive me to Mary's pajama party". After dropping her off, he exchanges the Caddie for the delivery truck and once more returns to his job of delivering pizzas.

Do you see the advantage of the flare? If there were no way to alert him if and when he was needed, he might be told to check out the park bench after each and every pizza was delivered. Since the brown paper bag would not often contain a new job, a lot of time would be wasted and the pizzas would get cold.

Returning to the old job is not an essential feature of the interrupt process. The old job could likely be discontinued and this will be determined by the new job instructions. The important thing about interrupts is that a capability to return to the old job exists, if it is needed.

Let's look at an actual example of interrupt use in the Home Computer. Some other models of computers require that the console be turned off when inserting modules, not so with the 99/4. You can be smack in the middle of a program when you yank out one module and stick in another.

Module insertion sends up an interrupt request flare that is nicknamed "Reset". Reset happens to be the highest priority interrupt recognized by the TMS 9900 microprocessor (more on priorities later). When the module is inserted in the port, the processor receives the reset signal, which tells it to fetch the two words of memory at addresses >0000 and >0002.

The first address contains the memory location of what will be register 0 of the reset routine's workspace. This is true for all 990 family microprocessors whether they are used in the Home Computer or in the International Users-Group $70,000 mainframe computer. This changing of workspaces is like the change over from the truck to the Cadillac in the pizza analogy. The workspace of the old job is saved so that its data will be preserved should you wish to return when the interrupt routine is completed.

Tom Hall, Sysop of a Edmonton TI BBS and member of the Edmonton Users Group, recently posted the following message on the TimeLine Unet TIN section. I think this aptly relates to Charles Carlson's 'Smorgas Board' and hopefully the hint will be recognized.

> In the absence of any really significant developments on the TI horizon, and the fact that the computer is a dinosaur merely waiting for the right peat bog to vanish into, it appears that our loyal brethren in the east, out of sheer desperation, it would seem, have decided to "bite the hand that feeds" them.
>
> The fact that the fanatic loyalty of a small number of people occasionally produces a brilliant hardware or software innovation can forestall but for a time the inescapable conclusion that, no matter how good a computer the TI 99/4A is (debatable on any number of points), it is only a matter of time before "the cutting edge" of technology sends the machine into the mists of microchip paleontology. In other words, dear friends, 10 years from now it is doubtful that any but a small handful of people will even know what a "TI 99/4A" was..........
>
> The moral of this little homily is simply this: neither TIMELINE (nor, for that matter, COMPUSERVE) were founded for the purpose of perpetuating a computer which neither CorComp, nor Myarc, nor any dozen other companies, however well-intentioned, can prevent from following other entities, both silicon- and carbon-based, into oblivion.
>
> In other words, dear friends, if TIMELINE is boring, or too expensive, or whatever, then YOU HAVE ONLY YOURSELVES TO BLAME! Just thank God that there are a few numbers you can dial with your little modems which provide you with something other than a busy signal!

Bob Boone recently posted a message on TimeLine's UNET TIN section saying that Bruce Caron's Disk Manager 1000 program is being beta tested for version 3.0. Version 3.0 is said to run E/A program image files, keep track of total sectors copied; deleted and moved in the File utility, fixes a early bug found in the DV/DF-80 file manipulation, and some less visible enhancements. We should have it real soon now!

The Hamilton Users Group (Channel 99) are beating the Ottawa Users Group to the punch by announcing a TI Fest to be held on February 22, 1986 at the Textile Centre (545 Main St E.) in Hamilton, Ontario. Run time is from 10 AM to 4 PM. Admission is a paltry two dollars. They are supposed to feature demonstrations of all kinds, workshops on various subjects, retail sales by various vendors in attendance, snack bar and a raffle for a MBX system. Be there as they say on NBC!

Scuttlebutt says Lou Phillips, Myarc, has PROMISED to have a WORKING new computer at the New Jersey User Group's Faire in late March and Bruce Ryan in Haliburton Ontario is negotiating with a company called MEGATRONICS in Germany to bring a least two exciting products for TI to North America namely being their enhanced XB module and stand alone 128K card.

Since posting our address in Computer Shopper's Users Group Section, numerous companies have sent us product announcement literature. The latest being Asgard Software of Rockville Maryland. The summation of their goodies is as follows:

> Disk Data Base is, simply put, the best disk file managing software available for the TI-99/4A. This program allows the user to catalog, with assembly language speed, up to 12,500 files in one single database. DDB has more features than any of it's aged competitors as well, allowing the user to sort the database by filename, diskname or to even print it out unsorted. The user also can selectively print the database in blocks of 250 entries at once, thus saving literally reams of paper by not printing unneeded catalogs. You can even convert catalogs made with other popular disk cataloging programs for use with DDB, so you can save your investment in time. DDB contains numerous assembly language subroutines for speedy sorts and snappy screens, and contains such user-friendly features as an on-line "HELP" files, prompts in English and not Computerese, one-touch menues, and more. This program is as useful with a small catalog as with a large since DDB will allow you to expand your old catalogs to fit any available new disk space - you could even put it on a hard disk drive! Disk Data Base requires Extended BASIC, a 32K memory expansion device, and a single disk drive and controlle , a second drive and a printer are recommended for optimum performance. Disk Data Base is an excellent value at $15.00, and as even better one in that it comes unprotected. No data monster is too large to be tamed by DDB, so isn't it time you tamed yours?
>
> GRAPHX is the most versatile graphics programs for the TI-99/4A, and GRAPHX Companion is your only partner for GRAPHX. This three disk package is a fine collection of new fonts, clipart, animation sequences, and pictures for use with GRAPHX. Each piece is finely detailed, carefully drawn, and artfully presented. The GRAPHX Companion user can feel free to use any of the included material within his her own works, either commercial or private. Asgard Software only asks that the source is acknowledged. GRAPHX Companion requires GRAPHX (available through us for $60), and the equipment detailed in the GRAPHX manual. The most useful complement to your GRAPHX drawing system has a suggested retail price of $10.00.
>
> Now, from the makers of the highly acclaimed GRAPHX Companion (see review in the Sept. '85 issue of MICROpendium) a companion product for the latest advance in 99/4A graphics, TI-Artist. All others aside, only our companion features art work from some of the best artists in the TI-99/4A world - Warren Agee, Chris Bobbitt, and many more fine craftsmen. Only our companion features not only the best art for GRAPHX (including the winner of the MICROpendium graphics contest) but also a collection of all new art work designed specifically for TI-Artist. Finally, our companion is also the least expensive available, only $7.00 for a two-disk package. Requires TI-Artist Version 2.0.
>
> Now available, the ultimate kitchen aid for the TI-99/4A! This useful utility for home-maker and chef alike is written entirely in the Forth language. Not only is it blindingly fast, it is also easy-to-use with one-touch menus and full prompts in English, not computer-ese. Its many powerful functions allow easy recipe entry and editing, printout and organization. Included are options for easily converting recipes to different serving sizes, and for printing stored recipes on index cards as well as regular paper. Ideal for maintaining a collection or recipes without messy, unorganized pen and paper! Requires Extended BASIC, 32K and a disk drive. Price $12.00.
>
> At last, an easy-to-use automatic telephone dialing program designed for use with popular terminal emulator programs such as FAST-TERM and P-TERM! Store up to 100 names and numbers in a single directory file for dialing at the touch of a single key. Features automatic re-dialing if a number is busy, full name and number editing options as well as complete print and display functions. The package includes a pre-written version for use with FAST-TERM, and easily modified generic version for other TE programs, and extensive documentation. Great for use with Compuserve, Source, and local BBS systems. Requires Extended BASIC, 32K, disk and a modem. Price $10.00.
>
> In order to purchase the above products, please write to: Asgard Software POB 10306 Rockville Maryland 20850.

That's all for this month. We'll see what the lions bring back to the den in March!

PROGRAMMING HELP FILE:

The purpose of this column is to present, to the user, techniques that will be useful in the writing of programs for the TI-99/4A home computer. If you can provide some prgramming insight that might be useful to someone, please, feel free to pass it on to me, and I'll get it into the next newsletter.

BASIC/EX-BASIC:

This month I will discuss the use of GOSUB and SUB statements.

When one programs there is often parts of the program that are used in several places yet perform the same operation. For example you may have a program where you are asked to press the enter key, when you are ready, at several points. The problem with this repetition of program statements is that they take up alot of space. To reduce the number of *program lines* that are repeated, and to save on the amount of memory that your program occupies you would use what are called subroutines.

Subroutines are basically programs within programs. Subroutines are sequences of program statements that may be executed from different lines within a program. IE- The SAME set of statements can be called upon to execute from different locations. The difference between a subroutine and a jump, branch, or GOTO (basicly these are all the same) is that the program will continue from the statement after the one where the subroutine call is made. A GOTO alters the flow of the execution of program statements in a manner in which you don't know where the next statement to be executed will be. GOTO's also allow you no way of knowing were, in the program, the GOTO jump was made, so that to get back to the line from which the program came EXTRA program statements have to be written. Subroutines eliminate all of this extra programming.

To use a GOSUB you must create your subroutine by placing its statements at some location away from the body of the *main program*, and terminate the GOSUB subroutine with a RETURN statement. The GOSUB subroutine is called by refering to the line number of the starting statement of the group to be executed. When the RETURN statement is encountered the program will continue executing from the statement immediately following the GOSUB statement that called the subroutine.

For example; 110 GOSUB 1000 when it is encountered within a program will cause the program to jump to line 1000 and continue executing until a RETURN statement is encountered. When the return statement is encountered the program will go back to the line immediately following line 110 and continue its execution from there.

There are several points about GOSUBs that should be mentioned.

1. GOSUB statements may be part of a GOSUB subroutine. This is called subroutine nesting.
2. All regular statements may be part of a GOSUB routine.
3. There may be more than one RETURN statement in a GOSUB routine.
4. Since a GOSUB statement refers to subroutine by a line number you may jump to any of the lines that exist in the subroutine, as long as a RETURN statement will be encountered at some point after the jump.
5. All variables used in the main program can be used by the GOSUB subroutine.

As you can see a GOSUB is a very powerfull operation. The GOSUB is made even more versatile by means of the ON GOSUB statement. With the ON GOSUB you can select one of many subroutines to go to. An example of an ON GOSUB is:

```
ON (KEY-48) GOSUB 100,200,300,400,500,600,700,800,900,1000
```

The SUB statement also defines a subroutine, but when the subroutine is referenced it is called by name. The are many subroutines that are built into TI-BASIC that are called in the same manner as a SUB subroutine. The only difference between the builtin subroutines and yours is that your subroutine was written by you. A subroutine call is made as follows:

```
CALL MYSUB(A,B)
```

The name MYSUB is used to refer to the subroutine and the variables A, and B within the parenthises are called parameters. Parameters are values, or variables, passed to a subroutine to use in its processing. Variables used as parameters can also hold data that is to be returned from a subroutine.

SUB subroutines are placed after the last end statement in your program. A typical SUB subroutine looks like the following:

```
1000 SUB TITLE(ROW,COL)
1001 rem  other statements would be here
1009 SUBEND
```

The subroutine must start with a SUB statement and end with a SUBEND statement. Another statement that is only allowed in a SUB is the SUBEXIT statement. The SUBEXIT statement is very much like a RETUPN statement since there may be more than one of them within a SUB, and when encountered will cause execution to return to the statement following the SUB call statement. If there is no SUBEXIT the SUB will return when it encounters the SUBEND.

All CALL statements to a SUB must have the correct type, and number of parameters matching the type and number of the parameters in the SUB statement, when executed, otherwise, an error will occur. A SUB may have no parameters if you wish.

To sum things up, the GOSUB and SUB subroutines may be used to simplify your prrograms and reduce replication of program statements thereby reducing the amount of memory that your program will use. This is important when you are writing large programs.

ASSEMBLY:

This month I will take a quick look at what registers are all about.

What is a register, and what are they used for?

Registers are places where bytes and words (pairs of bytes, in the 9900's case) can be manipulated quickly and simply. In the case of the 9900 CPU used in the TI, there are 16, 16 bit(word) general purpose registers. These registers are numbered 0 to 15, and are located in some area of the computer's memory. The register portion of a assembly language program is called the program's workspace area.

There are three other registers that are important to assembly programs. These registers are the Program Counter, Program Workspace, and Program Status registers, and are located within the 9900 CPU chip itself. (I will refer to this chip simply as the 9900, from here on.) These registers are word registers also.

The Program Counter register (PC) is used to keep track of where in a assembly program execution is currently at. The contents of this register are used pretty much like the line numbers that you have in Basic, the only difference being that it refers to an address of computer memory, where a machine instruction resides.

The Program Workspace register (WP) holds the memory address of the start of a assembly program's workspace area (the general purpose registers mentioned before).

The Program Status register (SR) is used to recognize error conditions that may arise in an assembly program. It records information about arithmetic operations and data value comparisions that are frequently performed in an assembly program. Each arithmetic operation or data comparision affects different bits of the status register and those assembly operations should be studied individually to understand their effects on the status register. I will leave that for a later discussion or for you to discover on your own.

The general purpose registers are used in many assembly operations. Most assembly instructions can use these registers as temporary locations for data. Since the location of the general purpose registers are always known to the CPU, there is a minimal amount of looking needed to be done to find a value in a register, and because of this registers are very fast for the CPU to use.

Registers are refered to by their number (0-15). If you use the R option when you assemble your program you may refer to a register by preceding the register's number by an 'R', ie - R0, R1, ...,R14, R15. There are special instructions that work with the three CPU registers, but this is not the time to go into their details.

Of the general purpose registers R0, R11, R12, R13, R14, and R15, have special purposes, in a few instructions. In particular registers 11 through 15 should not be used for general data manipulation.

That about sums up what registers are and what they are used for. For a better understanding of how registers are used you will have to learn more about the actual operation of each of the 69 assembly instructions, as well as the various addressing formats that those instructions may use. Part of the fun of assembly programming is learning new things about how the inside of your computer works. Happy programming!

FORTH:

The following article is reprinted from the May 1985 issue of the TINS newsletter. This article tells how to custom design your TI-Forth system loader.

Forth to You Too! by Lutz Winkler

As mentioned there is an elegant way to autoboot whatever you want your system to do, but before we can proceed with that we'll have to consider the following:

Since FORTH is a disk-based system it occupies memory which otherwise would be available for programming. That - in my opinion - is the reason TI provided many of the utilities as LOAD options. Look at the menu and also appendix F. Some of the options, ie- the editor, are essential, others are rarely needed. For instance, if you are not programming in code there is no need to clutter up memory with -CODE, and -ASSEMBLER. Similarly, if you aren't going to operate with graphics then there is no need for -VDPMODES, etc. It is not very likely that you will run out of memory while still in the learning process but why boot unnecessary stuff? I consider; only -DUMP, -COPY and -PRINT along with the editor is essential. To show you how fast memory is occupied even with your extra 32K, do this (assuming you are in FORTH): enter

```
            : FREE SP HERE - . ;
      (color FREE SP HERE minus dot semicolon)
```

Now enter FREE. You should get an answer of about 14140 (9790 if you opted for the 64-column editor). If you want to see how fast memory shrinks with each LOAD option boot a few more, but enter FREE between them. (If you are convinced enter COLD.) Here is what I autoboot and why:

1. -PRINT so I can list the screens I am working on;
2. -COPY so I can copy disks and screens;
3. -DUMP allows me to look at the parameter stack; and
4. -BSAVE a must to enable the quick autoboot

Recall that if you will be using a printer that you must change the device name on line #4 of screen #72 to reflect the one used by your system. ie. the sytem disk is set up to print to " RS232.BA=9600", if you use the parallel port then you must change this to " PIO", remembering the space at the start of the device name. To test your change enter -PRINT, turn on your printer and then enter:

```
      SWCH ." THIS IS A TEST" CR UNSWCH
```

A space is also needed after the ." in order for this message to be printed correctly.

Before we proceed with the actual set-up for your autoboot take a quick look at any screen between 8 and 19, no not in the manual, or your display (remember nn EDIT). Not much there that is legible, but believe it or not on those few screens resides every FORTH word that is identified in the Glossary of Resident words, only they are saved in a binary form. We will do the same with the LOAD options you decide upon by the use of BSAVE. So let's go. First, start off with COLD, then boot your options by entering the appropriate words (-PRINT, etc) and as the final one -BSAVE. Find the apostrophe key (Function O - that's "Oh" not "zero"). This is also a FORTH word pronounced TICK (page 3, Glossary). Now enter:

```
      ' TASK 22 BSAVE .
    (tick TASK 22 BSAVE dot)
```

Here is what's happening:

We are saving in binary form all that has been added to the dictionary (by booting the LOAD options) starting at screen 22. We can afford to wipe out 22 and some of the following screens because they contain the 64 column editor which you have either booted already (so it's in the autoboot dictionary) or you aren't going to use it. The final dot will print on your display the first screen after the BSAVE is done. All other LOAD options remain intact and can be booted when needed.

Now for the finishing touches. Enter EMPTY-BUFFERS 3 EDIT and carefully erase all but lines 0,1,2,13, and 15.

Line 2 - take out the parenthisis around 84 LOAD, and change 20 LOAD to read 22 BLOAD. You might want to replace the word BOOTING on line 0 with some other phrase which would let you know that you are using your new system-disk.

Line 5 - put: 0 DISK_LO ! and depending on how many drives you have, and wetner they are single or double sided enter one of the following: (for single sided drive skip this)

```
90 DISK_SIZE ! 180 DISK_HI ! (2 single sided drives)
180 DISK_SIZE ! 180 DISK_HI ! (1 double sided drive )
```

180 DISK_SIZE ! 360 DISK_HI ! (2 double sided drives)

Note that these words use the underline, not the hyphen. (If you have double density drives it is not quite that easy, you have to make several modifications to screens 33 and 40 as well as define a new word to install a proper disk header.)

After you have FLUSHed your edit go COLD and with any luck you will have a working system disk. You are ready for FORTH.

[He then goes on to say that you should read ch1 and ch2 of 'Starting Forth'. He also says that he used Appendix C (notes on 'Starting Forth') to make changes to his copy of the book. ed.]

CURIOSITIES AND PASTIMES

This column features a monthly BRAIN TWISTER for your intellectual entertainment. This month's twister is more of a massage than previous contents of this section have been. Hope everyone enjoys it!

```
Y Y E R A W D R A H S X I M Y J V H Z E
F B D N W L F C E Y E P S O D K R P P T
K M K D M W W A X Y H R R V C K Z I A T
P V P K D T Z S U D T I K C C X T N Z E
O N M X T R M S P P R N C C G O S U B K
S M E M O R Y E M O O T O O C J L H U S
O B D X E F R T F S F E L N X B E U J I
F V O T V I R T U S N R A S T J G Y C D
T D M B B R N E Z C U Y H O V D R V M Y
W H H G L M D R M J D U P L R Q A J A T
A L H Z E W J B U D Q R L E D I E T G E
R V S E M A G C M T K O A U A S O H A P
E Y C L X R D S P T E H P O A Z I R K O
M Z P E F E U B V I F R G B B W M I E A
V L I S T Q O F M F G B A U P Y I H I V
H F T T K A E V I U R T U Y B V E V P O
M O D U L E R Q R S A X S C K E R K E J
M A R G O R P A B D S N E G T X D C T V
U U I P O X G M C C T C R C C M V O I H
A F G P P H E L B M E S S A E P O E I Z
```

DATA
DATABASE
DISKETTE
SOFTWARE
PRINTER
MODULE
MODEM
GOSUB
LIST
DEBUG
FORTH
GAMES

KEYBOARD
MEMORY
HARDWARE
FIRMWARE
CONSOLE
CASSETTE
PROGRAM
RETURN
ALPHALOCK
ASSEMBLE
USER

NEWSLETTER EDITER
WINNIPEG 99/4 USERS GROUP
P.O.B. 1715
WINNIPEG, MANITOBA
CANADA, R3C 2Z6

EDMONTON TIERS
PO BOX 11983
EDMONTON ALBERTA
T5J 3L1