OFFICIAL NEWSLETTER OF THE
TIDEWATER 99/4A USERS GROUP INC.
4168 S. MILITARY HWY. LOT #21
CHESAPEAKE, VA. 23321

MAY 1988

A Non-Profit Virginia Corporation
dedicated to educating and
enlightening TI-99/4 users
to the full potential
of home computing.

Dues FREE
DALLAS TI HOME COMPUTER GROUP
P. O. Box 29863
Dallas TX 75229

OFFICERS:

PRESIDENT ---------------------------- Billy Denny 420-9631
VICE PRESIDENT Adminstration ---------- Allen Leibrand 485-5809
VICE PRESIDENT Operations ------------ Mike Couture 480-3943
SECRETARY TREASURE ------------------- Dick Hanson 587-1949
LIBRARIAN -------------- Mac and Cathy MacAllister 877-4699

****

MEETING NOTICE

The club meets every first and third tuesday of each month at E.C.P.I. (Electrinic Computer Programing Institute) located at 5555 Greenwich road in Virginia Beach. Educational classes start at 6:30 pm. followed by the regular meeting and discussion groups at 7:30 pm. Please note these times and dates on your calender.

There is another group meeting on the peninsula on the second tuesday of each month in room 101 at Warwick High School located at 51 Copeland Lane in Newport News. Formal meeting s begin at 7:30 pm with informal discussions before and after the meeting. Please note these times and dates on your calender.

NOTES FROM THE EDITOR

Hi everybody! Believe it or not, we have been getting a little fan mail. I recieved a coupla' local letters and one from California. If you have a comment, sent it to the return address shown on this newsletter.

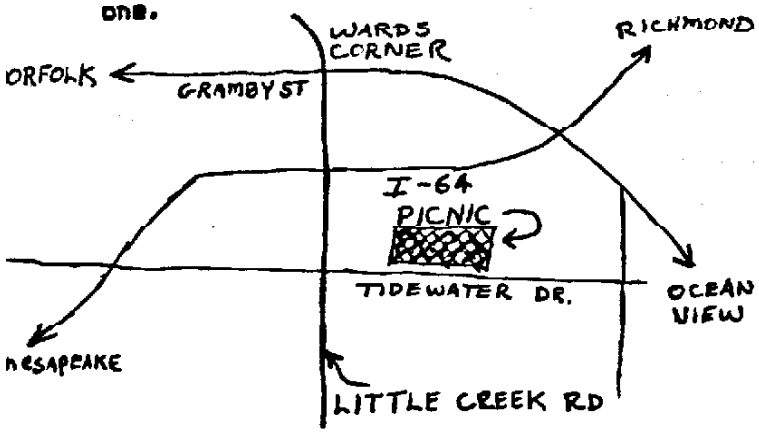
In this issue: Ken Woodcock's series for Extended Basic.
My own series "Beginners in Assembly Language.
Brian Combs "Not an Ordinary Article!".
And more!

Note: When sending stuff to the newsletter, please put "4168 S. Military Hwy. #21" on it. I live in a trailer park and the mailman won't know which box to put the mail in. Thanks, Allen.

@@@

Club Notes

Plans for the 2nd Annual Club Pinic are now finalized. The date is June 5th, from 1 to 4pm. Location Northside Park, on Tidewater Drive. Bring your own food and drink. For more info, call any club officer. The numbers are listed one page one.



Our own instructor, Mike Couture, is looking for suggestions for the lessons given at 6:30. Possible ideas are reviewing some of the programs in the library, Multiplan, PR-Base, Telco, and T.I. Writer/ Funlwriter. If you have any ideas, give Mike a call.

Not an Ordinary Article!
by Brian Combs

I am twelve year old boy that attends the meetings. If you are wondering who I, my name is Brian Combs (shortest one at the meetings). I would like to thank every one in the meetings and I mean everyone. They are so helpful if you have a problem.

For instance, I had a problem on a BRAPHX program and I got the help I needed and my parents had problems on a lot of programs and they got help. When I had a problem on BRAPHX they didn't treat me like a little kid. They treated me like an adult.

If you ever have any problems on your computer, ask anyone at the meetings and you will get the help you need.

My family (Mr & Mrs. Armstrong) and I would like to thank everyone at the user group.

Brian Combs

Wefax Update
by The Three Amigos

The latest is I am starting all over again. (back to the drawing board). It seems there are too many drawbacks to using the joystick port for interfacing the radio and timing control.

When using the j-port, it renders the keyboard partially unusable. This is very annoying because it hinders future expansion.

Also the design of my interface employes timing counters to convert frenquency to digital. This has an inherent flaw. It overlaps if you miss-tune your radio.

So I am working on a new and inproved design. Mike Couture has worked out the bugs in a Phase Lock Loop circuit which effectly converts the frequency shifts to an analog voltage. This could be fed to one of the A/D ports of a MBP board. The board will enable the computer to read the voltage and make a picture. Timing will be maintained by the clock chip on the MBP card.

A spinoff of this proceedure will be feeding the Most Significant Digit from the A/D chip in the MBP card to the RS232 #2 port and RTTY reception becomes possible.

Keep in mind this work is still on the drawing board and may take a year before finished. More next issue.

-Allen

Helpful References:

RTTY TODAY:
A Modern Guide To Amateur Radioteletype.
by Dave Ingram

HARDWARE MANUAL FOR THE T.I.99/4A.
by Michael Bunyard,PE

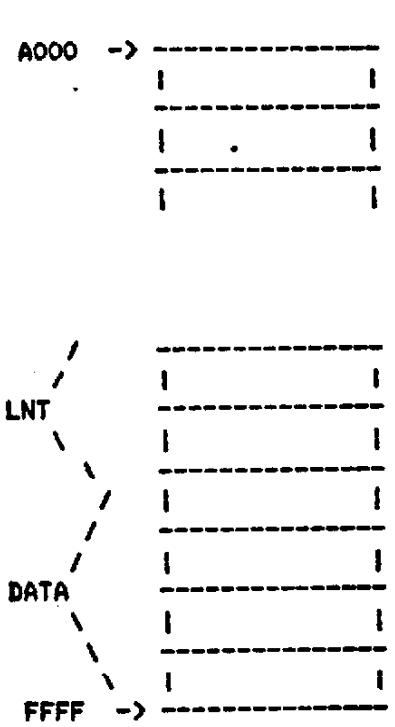
9900 FAMILY SYSTEMS DESIGN
by Texas Instruments.

ANALYZING a BASIC PROGRAM (part A)
by Ken Woodcock

Have you ever wondered about how your computer stores all the instructions that you enter when you type in a BASIC (or XBASIC) program? If not then don't waste your time reading this article. Personally, I am the curious type, so I set out to try to find out how programs are stored and what happens when they are run. I discovered some interesting (at least to me) things. I don't pretend to know all the details but will share with you what I have found. This discussion applies to both console BASIC and EXTENDED BASIC. For this discussion, try to visualize the memory area as a stack of bytes with the lowest numbered byte on top and extending to the highest numbered byte on the bottom.

Lets begin at the point where you've just turned your system on and selected TI BASIC. The cursor is sitting there flashing. . . .waiting. You are in the command mode. You can type in anything you want to but when you press ENTER your computer is going to check it for validity. It expects either a NUMBER or a COMMAND first. If it is a number, it "thinks" you are entering a program line with that number as the line number. If it is a command, it tries to execute the entire line you typed in. In either case the line will be checked for proper structure (SYNTAX) and an error message issued if something is wrong. If the first word was not a number or a valid command, an error message will be issued also. Let's follow the course of the program line entry. When ENTER is pressed the line of code is searched for RESERVED WORDS. These are commands or statements which have special meaning to the BASIC INTERPRETER (ie. PRINT,IF,THEN). Each of these is replaced with a single byte called a TOKEN. This is called "tokenizing" or "crunching". Now the data is moved into the memory area which is reserved for it's use. If you have the 32K memory expansion and are using EXTENDED BASIC, this will be the "HIGH MEMORY" area from A000 to FFFF otherwise it will be VDP memory (the 16K in the console). In either case the actual data will be inserted at the highest free address (ie. FFFF in High Mem). In addition to the "crunched" data, two other bytes are added to each line. One byte at the begining is set equal to the length of the "crunched" line. This byte is apparently only used when listing the program because it can be changed to any value and not affect the operation of the program. Some software producers deliberately set this byte to zero to prevent listing of the program. The second byte is added to the end of each line as an "end of line marker". It is always 00. The line number information is stored separately from the program data in a "Line Number Table" (LNT). The LNT is stored immediately above the program and is pushed along as new lines are entered. There are 4 bytes for each entry in the LNT ; two bytes for the line number and two bytes which point to the start of the line data. The LNT is always sorted in line number sequence(from highest to lowest), unlike the program data area which is kept in "as entered" order. There are two important addresses which deal with the LNT. These are >8332 which points to the end of the LNT and >8330 which points to the start (highest line #). These are in "scratch pad ram". There are other pointers in this area which are used by the Basic interpreter while the program is running. We will get to them later. The following 1 line program can be used to inspect your LNT and print the line # and address of the line data for each entry. It can be typed in from the command mode of XB and therefore will not interfere with the program you are trying to inspect.

```
CALL PEEK(-31952,A,B,C,D)::FOR
X=C*256+D-65539 TO A*256+B-65536 STEP
-4::CALL PEEK(X,E,F,G,H) ::PRINT
"LINE#";E*256+F;" ADD";G*256+H::NEXT
X
```



Introduction to Assembly Language.
Part 2 by Allen Leibrand

DEF START

Def is a assembler command that tells the assembler to put the label "START" and the address it represents into the ref/def table located in CPU memory >3FFF to >3000

This is done so the program may be accessed from the editor/assembler loader as a program name or from basic using CALL LINK.

REF VSBW

REF statments are similiar to to DEF statements except the refer to subroutines that will be loaded later at run time- in this case VSBW is a subroutine that is loaded by the EDITOR/ASSEMBLER loader thru call init.

STATUS EQU >837C

STATUS is a label used later in the program. EQU is short for "equate" or equal. >837C is the location of the status byte. The status byte is a register that sets certain bits some operations such as compare, move, darlink, and others.

These bits will affect conditional jumps such as JEQ (jump if equal), JLT(jump less than) and others. For now don't worry about it.

WS BSS 32

WS is a label. BSS means BLOCK STARTING with SYMBOL. This means starting at label "WS" we will reserve 32 bytes of memory. This will be used later in the program for our workspace.

SAVE BSS 2

"SAVE" is a label. This instruction will reserve 2 bytes of memory starting at the label "SAVE". This will be used to save the return address for the calling program, in this case BASIC.

START MOV R11,@SAVE

This is where the program starts running. The label "START" was defined in the program for external use.

The "MOV" takes the number stored at register 11 and puts the data at label "SAVE". Register is where the the computer stores the return address of the callind program. This will be used later to return to basic.

LWPI WS

Load workspace pointer immediate to label "WS". At WS we set aside 32 bytes of memory. This was done so that data at the original workspace is not disturbed. If it was it could crash the basic program.

LI R1,>2000

Load Immediate Register 1 with Hex 2000. When we clear the screen, we will write >20 to all positions on the screen image table. This command is for preparing to use the VSBW routine.

CLR R0

The Screen Image Table starts at >0000 in VDP memory. R0 is used by the VSBW routine to indicate where in VDP memory you wish to write the data. The data to be written is stored in R1.

LOOP BLWP @VSBW

"LOOP" is a label to be used to return to this point again in the program. Branch and Load Workspace Pointer is a command used to link to subroutines which have their own workspaces. In this case the subroutine is "VSBW".

```
        INC  R0
```

Increment R0 by one. This is done to set up for the next write to the screen.

```
        CI   R0,768
```

Compare Immediate R0 to 768. This instruction compares the value of R0 with 768. If R0 is less than 768, the less than bit is set in the status register. If they are equal, the equal bit will be set.

```
        JNE  LOOP
```

Jump Not Equal. This instruction will check the status register. If the equal bit is not set it will jump to the label "LOOP". If it is set, the computer will continue to thi next instruction.

```
OUT     LWPI >83E0
```

"OUT"is a label. It serves no real purpose except as a programmers aid that tells us this part of the program deals with returning to basic.

Load Workspace Immediate to >83E0 is loading the basic programs workspace.

```
        MOV  @SAVE,R11
```

We now return the return address to R11.

```
        CLR  @STATUS
```

Clear the memory at the label "STATUS". BASIC will check tha status register on return.

```
        B    *R11
```

Branch to the address in R11. This orders the computer to return to the BASIC program.

```
        END
```

Tells the assembler to stop assembling.

Enclosure:

```
TI BASIC W/ E/A INSTALLED
100 CALL INIT
110 CALL LOAD("DSK1.CLEAR")
120 CALL LINK("START")
130 END

ASSEMBLY CODE

       DEF  START
       REF  VSBW

STATUS EQU  >837C
WS     BSS  32
SAVE   BSS  2
START  MOV  R11,@SAVE
       LWPI WS
       LI   R1,>2000
       CLR  R0
LOOP   BLWP @VSBW
       INC  R0
       CI   R0,768
       JNE  LOOP
OUT    LWPI >83E0
       MOV  @SAVE,R11
       CLR  @STATUS
       B    *R11
       END
```

Contest of the Month

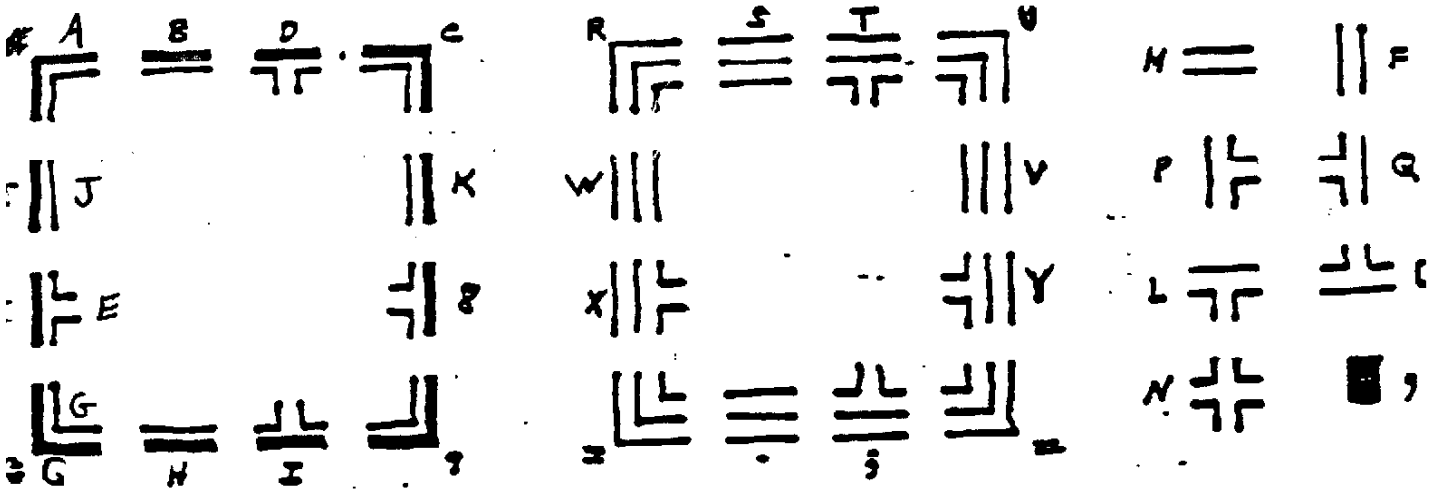
Can you correctly identify what the above person is doing? If you can send in your answer to the newsletter. The first person to do this will win a box of ten disks.

The address: Contest of the Month
c/o Allen Leibrand
4168 S. Military Hwy.
Lot #21
Chesapeake, VA. 23321

Good Luck!

PRBASE Reference Chart for "CREATE" subprogram
by Vic Schaffner via LA 99ers TOPICS reformatted by Mary Phillips, Ozark 99ers

Press <CTRL> and your letter choice to make graphic character shown.
These characters are for designing your data screen and are not printed.



Press <FCTN> key
and this key to get this operation:

- E ——— UP CURSOR
- S ——— LEFT CURSOR
- D ——— RIGHT CURSOR
- X ——— DOWN CURSOR
- 1 ——— Delete character under cursor
- 2 ——— Insert space under cursor
- 3 ——DANGER—— Erase complete screen
- 4 ——— Half output to PIO or RS232
- 6 ——— PROCEED after screen design is finished.
- 9 ——DANGER—— Return to CREATE title screen. NO SCREEN SAVED

<ENTER> ——— Moves cursor to beginning of next line.

[ ] ——— All data entered between these is displayed as UPPER CASE

( ) ——— Data entered between these is displayed in UPPER and LOWER case.

SCREEN COLOR CHOICE - At "CREATE" title screen, press "F" for foreground press "B" for background

****

PRBASE Reference chart for "DATA MANAGEMENT" commands

A Add Record	L Print Labels	FCTN X Scroll to Next Screen
B Boot Data Base	N Go to Screen #	FCTN E Scroll to Previous Screen
C Control Codes	O Program options	FCTN D Next Alphabetical Screen
D Delete Record	P Print Screen	FCTN S Previous Alphabetical Screen
E Edit Record	Q Quit PRBASE	
F Find String	R Print Reports	CTRL X Rapid Scroll Screen 1 - End
G Global Search	S Sort Index	CTRL E Rapid Scroll Screen End - 1
H Help Commands	U Use Index to Find	CTRL D First Alphabetical Screen