West Jax 99er News

OCTOBER 1988

The WEST JAX 99'ERS is a non-profit computer users group for the TI-99/4A Home Computer. NOT affiliated in any way with Texas Instruments. The club's mailing address is PO BOX 176 Orange Park Florida 32067.

MEETINGS are held on the Second and Fourth Tuesday of each Month in the auditorium of the Webb Library. It is located two lights west of Blanding Boulevard on 103rd Street. The first meeting of the month is the Business meeting with workshop time after adjournment. The second meeting is strictly workshop time.

OFFICERS

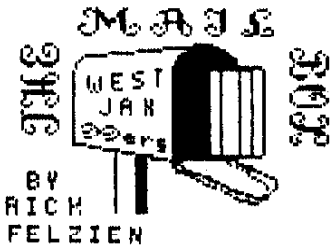
President...Rick Felzien.....(904) 772-9162
Treasurer...Thomas LeMay.....(904) 282-5220
Secretary...Ralph Glattli.....(904) 751-1308
Librarian.....Zach Ziegler.....(904) 389-2194

For newsletter suggestions and submissions, contact Rick Felzien.

This month we have the Mail Box and a revised version of Wordsearch by Richard Kotrba.

We also have our usual installment of the Basic Assembler by Steve Peacock.

I have included reprints of a couple good TI-Base tutorials from the Cleveland and the Paris newsletters.



SFV 99'er Times Sep 88
1. TI Trivia
2. Graphing

Ottawa 99'er Newsletter Sep 88
1. Letter to fairware authors
2. Fast Extended basic
3. TI Basic

Cleveland Area 99'ers Sep 88
1. Ramdisk tip
2. TI-Writer tips
3. Nine months pregnant?

Newsletter of Decatur Sep 88
1. TI-Base review
2. Group library listing

MAD/HUG newsletter Sep 88
1. Plans for a nice computer desk

N.O.V.A. 99'ers Sep 88
1. Font writer II revisited

Pug peripheral Sekp 88
1. Getting most from cassette
2. Multiplan article

Aloha U.G. newsletter Sep 88
1. An article on PLUS!.
2. TI-Base review

Greater Tampa Bay U.G. Aug 88
1. MAX-RLE doc file

ROM newsletter Aug 88
1. Assembly tutorial
2. And so FORTH #33
3. Assembly Inverse video prog.

Spirit of 99 Sep 88
1. Two joysticks in one

T H E B A S I C A S S E M B L E R #5 By Steve Peacock

USING SPRITES

This month is dedicated to SPRITES and how to use them in TI Assembly. In order to use sprites to their fullest, information must be written to three tables and one register. These tables are SPRITE ATTRIBUTE TABLE, SPRITE DESCRIPTOR TABLE, and SPRITE MOTION TABLE. The register is number one, of the write only registers. There are eight of these registers that can only be written to. Any character from ASCII 32 to ASCII 159 can be used for a sprite, in TI Assembly. In order to have a sprite move it MUST be character number 128 to 159 (dec) or >80 to >9F (hex). Sprite motion can be done with up to 32 different characters. These are numbered 0 to 31. It is a good habit to start using number 0 and work up. This will allow you to disable the higher numbers, and your program will run faster. The SPRITE ATTRIBUTE TABLE holds the Dot Row, Dot Column, Character Number, and Color Code (in that order). The SPRITE DESCRIPTOR TABLE holds the Definition of the sprite. If Magnify 3 or 4 is used, the definition is listed Top Left, Bottom Left, Top Right and Bottom Right (in that order). The *SPRITE MOTION* table holds the Verticle Motion and Horizontal Motion and two bytes of padding (in that order).

SPRITE ATRIBUTE		SPRITE MOTION		SPRITE DESCRIPTOR			
SPRITE #	ADDRESS	ADDRESS		CHAR #	(D H)	ADDRESS	
0	>0300	>0303	>0780 >0783	128	>80	>0400	>0407
1	>0304	>0307	>0784 >0787	129	>81	>0408	>040F
2	>0308	>030B	>0788 >078B	130	>82	>0410	>0417
3	>030C	>030F	>078C >078F	131	>83	>0418	>041F
4	>0310	>0313	>0790 >0793	132	>84	>0420	>0427
5	>0314	>0317	>0794 >0797	133	>85	>0428	>042F
6	>0318	>031B	>0798 >079B	134	>86	>0430	>0437
7	>031C	>031F	>079C >079F	135	>87	>0438	>043F
8	>0320	>0323	>07A0 >07A3	136	>88	>0440	>0447
9	>0324	>0327	>07A4 >07A7	137	>89	>0448	>044F
10	>0328	>032B	>07A8 >07AB	138	>8A	>0450	>0457
11	>032C	>032F	>07AC >07AF	139	>8B	>0458	>045F
12	>0330	>0333	>07B0 >07B3	140	>8C	>0460	>0467
13	>0334	>0337	>07B4 >07B7	141	>8D	>0468	>046F
14	>0338	>033B	>07B8 >07BB	142	>8E	>0470	>0477
15	>033C	>033F	>07BC >07BF	143	>8F	>0478	>047F
16	>0340	>0343	>07C0 >07C3	144	>90	>0480	>0487
17	>0344	>0347	>07C4 >07C7	145	>91	>0488	>048F
18	>0348	>034B	>07C8 >07CB	146	>92	>0490	>0497
19	>034C	>034F	>07CC >07CF	147	>93	>0498	>049F
20	>0350	>0353	>07D0 >07D3	148	>94	>04A0	>04A7
21	>0354	>0357	>07D4 >07D7	149	>95	>04A8	>04AF
22	>0358	>035B	>07D8 >07DB	150	>96	>04B0	>04B7
23	>035C	>035F	>07DC >07DF	151	>97	>04B8	>04BF
24	>0360	>0363	>07E0 >07E3	152	>98	>04C0	>04C7
25	>0364	>0367	>07E4 >07E7	153	>99	>04C8	>04CF
26	>0368	>036B	>07E8 >07EB	154	>9A	>04D0	>04D7
27	>036C	>036F	>07EC >07EF	155	>9B	>04D8	>04DF
28	>0370	>0373	>07F0 >07F3	156	>9C	>04E0	>04E7
29	>0374	>0377	>07F4 >07F7	157	>9D	>04E8	>04EF
30	>0378	>037B	>07F8 >07FB	158	>9E	>04F0	>04F7
31	>037C	>037F	>07FC >07FF	159	>9F	>04F8	>04FF

You can use any of these characters as any sprite number. Sprite number three does not have to be character number 782.

When using sprites you need to indicate how many are to be moving. The number of moving sprites is stored at address >837A. If you want nine sprites to move (#0 to #8) put a nine at this address. This is done with the following command:

```
    LI   R5,>0900
    MOVB R5,@>837A
```

Sprites #9 to #31 can still be on the screen, but can not move. It is also a good idea to turn these off. To do this write >D0 in the sprite attribute list of the first unused sprite:

```
    DATA >6080,>8304,>D000
```

This deleates all sprites above the character #>83.

When using the sprite motion table, the VDP INTERUPTS must be inabled and then disabled. ALWAYS disable the VDP interupts before accessing the VDP RAM. To do this, use the following commands:

```
    LIMI 2
    LIMI 0
```

Put this in a frequently used loop. This will also let you use the FCTN/= to return to the main title screen.

In setting magnification, VWTR number one is used. To set magnification, ONE >E0 is written to the register, TWO >E1, THREE >E2, FOUR >E3. For example:

```
    LI   R0,>01E2
    BLWP @VWTR
```

Will set magnification THREE. Magnification ONE is default and does not need to set, unless you have set 2, 3, or 4, and you want to return to 1.

In TI Assembly the speed of a sprite is controlled by the numbers >00, >01 to >7F and >FF to >81. In decimal this is 0, 1 to 127 and -1 to -127.

DEC	HEX		DEC	HEX
0	>00		0	>00
1	>01	SLOW	-1	>FF
2	>02		-2	>FE
.	...		..	...
126	>7E		-126	>82
127	>7F	FAST	-127	>81

The left most dot column, on the screen is numbered >00. The next is >01 up to >FF, being the right most column. The dot rows are a little different. The top row is numbered >FF the second is >00 with the bottom row (that is visible) being >BE.

```
ROW
>FF
>00      COLUMN
>01      >00,>01,>02,>03....>FD,>FE,>FF
>02
>03      The Center Point of the Screen is ROW >5F COLUMN >7F
...
>BD
>BE
```

```
*****************************************************************************
*
*PROGRAM BA5A==>Basic Assembler #5 Assembly Version
*USING SPRITES
*(C)1985 S. PEACOCK
*
*****************************************************************************
       REF  VWTR,VMBW     *VIDEO WRITE TO REG AND VIDEO MULTIPLE BYTE WRITE
       DEF  START         *DEFINE THE START OF PROGRAM (CAN BE ANY NAME)  .
START  LI   R5,>0200      *WILL BE USING ONLY TWO SPRITES #0 AND #1
       MOVB R5,@>837A     *STOPS MOTION OF SPRITES #2 THROUGH #31
**************************(NO EXTENDED BASIC PARALLEL)
       LI   R0,>01E1      *>01 WTR #1, E1 SETS MAGNIFICATION 2
       BLWP @VWTR         *WRITE THIS INFORMATION
       LI   R0,>0400      *ADDRESS FOR CHARACTER #>80 (128)
       LI   R1,SD1        *THE DEFINITION FOR THE CHARACTER
       LI   R2,8          *EIGHT BYTES TO WRITE
       BLWP @VMBW         *WRITES THE DATA LABELED DEF1
       LI   R0,>0408      *ADDRESS FOR CHARACTER #>81 (129)
       LI   R1,SD2        *THE DEFINITION FOR THE CHARACTER
       LI   R2,8          *EIGHT BYTES TO WRITE
       BLWP @VMBW         *WRITES THE DATA LABELED DEF2
       LI   R0,>0300      *ADDRESS OF SPRITE ATTRIBUTE FOR SPRITE #0
       LI   R1,SA1        *DATA TO WRITE DOT ROW, DOT COL, CHAR #, & COLOR
       LI   R2,4          *FOUR BYTES TO WRITE
       BLWP @VMBW
       LI   R0,>0304      *ADDRESS OF SPRITE ATTRIBUTE FOR SPRITE #1
       LI   R1,SA2        *DATA TO WRITE DOT ROW, DOT COL, CHAR #, & COLOR
       LI   R2,5          *THE 5th BYTE (>D0) TURNS OFF ALL SPRITES
**************************NUMBERED ABOVE >81 (NO EXTENDED BASIC PARALLEL)
       BLWP @VMBW
       LI   R0,>0780      *ADDRESS FOR SPRITE MOTION FOR SPRITE #0
       LI   R1,SM1        *DATA TO WRITE VERTICAL MOTION, HORIZONTAL MOTION
       LI   R2,4          *FOUR BYTES TO WRITE
       BLWP @VMBW
       LI   R0,>0784      *ADDRESS FOR SPRITE MOTION FOR SPRITE #1
       LI   R1,SM2        *DATA TO WRITE VERTICAL MOTION, HORIZONTAL MOTION
       LI   R2,4          *FOUR BYTES TO WRITE
       BLWP @VMBW
LO     LIMI 2             *ENABLE VDP INTERUPTS
       LIMI 0             *DISABLE VDP INTERUPTS
       JMP  LO            *UNCONDITIONAL JUMP BACK TO LO
SD1    DATA >AA55,>AA55,>AA55,>AA55  *DATA FOR DEFINITION OF SPRITE #0
SD2    DATA >FF81,>8181,>8181,>81FF  *DATA FOR DEFINITION OF SPRITE #1
SA1    DATA >5F7F,>800F              *DATA FOR ATTRIBUTS OF SPRITE #0
SA2    DATA >0101,>8106,>D000        *DATA FOR ATTRIBUTS OF SPRITE #1
SM1    DATA >0000,>0000  *DATA FOR MOTION OF SPRITE #0 (TWO BYTES PADDING)
SM2    DATA >05FB,>0000  *DATA FOR MOTION OF SPRITE #1 (TWO BYTES PADDING)
       END
```

WORDSEARCH

THIS PROGRAM APPEARED IN A NEWSLETTER LAST YEAR. I APOLOGIZE THAT I DO NOT KNOW WHICH NEWSLETTER IT APPEARED IT BUT L.P. THOMAS' NAME APPEARS IN THE PROGRAM AS HAVING EDITED THE PROGRAM. WHEN I TYPED IT THE PROGRAM IT WORKED PERFECTLY BUT ONLY FILLED A QUARTER OF A PAGE. THE PAGE HEADINGS ALSO NEEDED SOME MODIFICATIONS. THE PROGRAM WILL NOW FILL A FULL PAGE WITH EQUAL SPACING BETWEEN LETTERS. A SAMPLE PROGRAM IS INCLUDED AFTER THE LISTING. THIS IS A GREAAT PROGRAM TO PRODUCE INDIVIDUALIZED GAMES THAT CAN BE USED TO INCREASE VISUAL DISCRIMINATION BETWEEN RANDOM LETTERS AND WORDS FORMED IN EIGHT DIFFERENT DIRECTIONS. BY USING WORDS THAT HAVE A SPECIAL MEANING TO THE STUDENT, IT PROVIDES MOTIVATION TO FIND THE WORDS. IT CAN ALSO BE USED AS A PARTY GAME WITH INDIVIDUALS OR TEAMS COMPETING TO SOLVE THE PUZZLES. -

RICHARD KOTRBA

```
100 CALL CLEAR :: PRINT :"THIS PROGRAM GENERATES A" :: PRINT :"WORDSEARCH PUZZLE
." :: PRINT :"IT REQUIRES EXTENDED BASIC" :: PRINT :"IN ITS PRESENT FORM BUT" ::
 PRINT :"SHOULD RUN IN BASIC IF"
110 PRINT :"RE-EDITED.": : :: PRINT :"PRESS ENTER TO CONTINUE" :: INPUT G0$ :: C
ALL CLEAR :: PRINT "THERE ARE FOUR LINES TO" :: PRINT "CHECK FOR PRINTER OUTPUT"
 :: PRINT :"PRESENT SETUP IS FOR A        GEMINI-10X"
120 PRINT :"CHECK LINES:" :: PRINT " 1670 (FORM FEED)" :: PRINT " 1750!" :: PRIN
T "     )(PRINT STYLE)" :: PRINT " 1760!" :: PRINT " 1930 (OPEN 'PIO')" :: PRINT
 :"THESE INSTRUCTIONS ARE ON"
130 PRINT "LINES 10 THRU 60 SO THEY" :: PRINT "CAN BE REMOVED": :: PRINT :"PROGR
AM EDITED BY L.P.THOMAS" :: PRINT "FURTHER EDITING BY RICHARD  KOTRBA" :: PRINT
:"ENTER TO CONTINUE" :: INPUT G0$ :: CALL CLEAR
140 REM WORDSEARCH
150 DIM L$(26),N$(40),M$(20,20):: CALL CLEAR :: PRINT TAB(9);"WORDSEARCH": : : :
 PRINT "YOU MAY ENTER UP TO FORTY" :: PRINT :"WORDS. (OVER 30 GETS":"DIFFICULT.
) THEY WILL BE"
160 PRINT :"PLACED HORIZONTALLY," :: PRINT :"VERTICALLY, OR DIAGONALLY" :: PRINT
 :"IN A WORDSEARCH PUZZLE.": : : :: OPTION BASE 0 :: FOR C=1 TO 26 :: L$(C)=CHR$(64+C)::
 NEXT C
170 FOR A=1 TO 20 :: FOR C=1 TO 20 :: M$(A,C)="." :: NEXT C :: NEXT A
180 INPUT "HOW MANY WORDS?":W :: IF W<41 THEN 210
190 PRINT :"SORRY,40 OR LESS!": : :: GOTO 180
200 CALL CLEAR
210 PRINT " DOES THIS PUZZLE HAVE" :: INPUT " A THEME? Y/N ":NAM$ :: IF NAM$="N"
 THEN 240
220 IF NAM$<>"Y" THEN 210
230 INPUT "WHAT IS NAME? ":NAM$
240 FOR C=1 TO W :: PRINT :"NUMBER ";C;
250 INPUT "WORD: ":N$(C):: A=0 :: L=LEN(N$(C)):: IF L<20 THEN 270
260 PRINT :"SORRY,MUST BE SHORTER!": : :: GOTO 250
270 A=A+1 :: IF A<100 THEN 320
280 PRINT :"COULDN'T FIT WORD IN!" :: PRINT :"CHOOSE: 1 START ALL OVER" :: PRINT
 TAB(10);"2 TRY ANOTHER WORD" :: PRINT TAB(10);"3 STOP NOW" :: CALL SOUND(150,14
97,2)
290 CALL KEY(0,K,S):: IF (K<49)+(K>51) THEN 290
300 CALL CLEAR :: ON K-48 GOTO 170,250
310 W=C-1 :: GOTO 660
320 D=1 :: E=1 :: F=-1 :: RANDOMIZE :: IF RND<=.5 THEN 340
330 F=1
340 G=-1 :: IF RND<=.5 THEN 360
350 G=1
360 H=2 :: IF F<>1 THEN 380
370 D=0
```

```
380 IF G<>1 THEN 400
390 E=0
400 IF RND<.75 THEN 420
410 H=1
420 IF RND>.25 THEN 440
430 H=0
440 COL=20 :: ROW=20 :: IF H<>1 THEN 460
450 ROW=20-L
460 IF H<>0 THEN 480
470 COL=20-L
480 IF H<=1 THEN 500
490 ROW=20-L :: COL=20-L
500 IF COL<>20 THEN 520
510 E=0
520 IF ROW<>20 THEN 540
530 D=0
540 II=INT(ROW*RND+1):: IJ=INT(COL*RND+1):: I=II+D*L :: J=IJ+E*L :: IF H=1 THEN
620
550 IF H=0 THEN 590
560 FOR P=1 TO L :: IF M$(J+(P-1)*G,I+(P-1)*F)="." THEN 580
570 IF M$(J+(P-1)*G,I+(P-1)*F)<>SEG$(N$(C),P,1)THEN 270
580 NEXT P :: A=0 :: FOR P=1 TO L-1 :: M$(J+P*G,I+P*F)=SEG$(N$(C),P+1,1):: NEXT
P :: M$(J,I)=SEG$(N$(C),1,1):: GOTO 650
590 FOR P=1 TO L :: IF M$(J+(P-1)*G,I)="." THEN 610
600 IF M$(J+(P-1)*G,I)<>SEG$(N$(C),P,1)THEN 270
610 NEXT P :: A=0 :: FOR P=1 TO L-1 :: M$(J+P*G,I)=SEG$(N$(C),P+1,1):: NEXT P ::
 M$(J,I)=SEG$(N$(C),1,1):: GOTO 650
620 FOR P=1 TO L :: IF M$(J,I+(P-1)*F)="." THEN 640
630 IF M$(J,I+(P-1)*F)<>SEG$(N$(C),P,1)THEN 270
640 NEXT P :: A=0 :: FOR P=1 TO L-1 :: M$(J,I+P*F)=SEG$(N$(C),P+1,1):: NEXT P ::
 M$(J,I)=SEG$(N$(C),1,1)
650 NEXT C
660 CALL CLEAR
670 INPUT " PRINT ANSWER KEY? Y/N ":ANS$ :: IF ANS$="Y" THEN 720
680 IF ANS$<>"N" THEN 670
690 ANS$="N" :: PRINT :"ONE MOMENT PLEASE" :: GOSUB 910 :: FOR A=1 TO 20 :: FOR
B=1 TO 20 :: IF M$(A,B)<>"." THEN 710
700 M$(A,B)=L$(INT(26*RND+1))
710 NEXT B :: NEXT A :: CALL CLEAR :: FOR A=1 TO 20 :: FOR D=1 TO 20 :: PRINT M$
(A,B);:: NEXT B :: PRINT :: NEXT A
720 IF OPT<>0 THEN 740
730 GOSUB 980
740 GOSUB 890 :: INPUT "ADJUST PRINTER AND THEN      PRESS ENTER.":START$ :: FOR
 Y=1 TO X :: PRINT #1:"                          **********WORDSEARCH PUZZLE**********
*" :: IF ANS$="N" THEN 760
750 PRINT #1:"                    !!!!!!!!!!!!!!!ANSWER KEY!!!!!!!!!!!!!!!"
760 IF NAM$="N" THEN 780
770 PRINT #1:TAB(40-INT(LEN(NAM$)/2));NAM$
780 PRINT #1 :: FOR A=1 TO 20 :: PRINT #1:"            ";:: FOR B=1 TO 20 :: PRINT
 #1:M$(A,B);"  ";:: NEXT B :: PRINT #1 :: PRINT #1 :: NEXT A :: IF ANS$="N" THEN
 800
790 PRINT #1:CHR$(12);: GOTO 690
800 PRINT #1 :: LST=INT((W/2)+.6):: FOR C=1 TO LST :: IF C>9 THEN 820
810 T=20 :: GOTO 830
820 T=19
830 PRINT #1:TAB(T);C;"-";N$(C);:: IF LST+C>W THEN 880
840 IF LST+C>9 THEN 860
850 T=42 :: GOTO 870
860 T=41
870 PRINT #1:TAB(T);LST+C;"-";N$(LST+C)
```

```
880 NEXT C :: PRINT #1:CHR$(12):: NEXT Y :: CLOSE #1 :: END
890 PRINT : : :: INPUT "HOW MANY COPIES? ":X :: IF X<1 THEN 890
900 PRINT #1:CHR$(27);CHR$(66);CHR$(1):: PRINT #1:CHR$(27);CHR$(67):: RETURN
910 GP=W*1.5
920 GP=INT(GP/2):: IF GP=0 THEN 970
930 FOR IP=1 TO W-GP :: JP=IP
940 KP=JP+GP :: IF N$(JP)<=N$(KP)THEN 960
950 CG$=N$(JP):: N$(JP)=N$(KP):: N$(KP)=CG$ :: JP=JP-GP :: IF JP>0 THEN 940
960 NEXT IP :: GOTO 920
970 RETURN
980 OPEN #1:"PIO" :: OPT=1 :: RETURN
990 PRINT #1:"                    !!!!!!!!!!!!!!!!ANSWER KEY!!!!!!!!!!!!!!!!"
```

**********WORDSEARCH PUZZLE**********
!!!!!!!!!!!!!!!!ANSWER KEY!!!!!!!!!!!!!!!!

STATES

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

**********WORDSEARCH PUZZLE**********

STATES

H H X C P X K Q P P J S A K W G A B M G

H D W A A J B Q U A I N I G R I V X G F

P K P G G N I S N O C S I W N M A H E W

V B W N G I R T I P N S M D W N C N O B

Q C K P D I A R A O C E M H O A O A R O

Q G N A S O J Z A P L H F Z L G W P G Z

Y B H P P N G T D F V T I I E O M V I O

K O P W D W E F A S D R F R I O B W A C

C A V J Y K L H V V A O Q K N G A G G I

U K J Z G O Z J E S R V W T C S J R L X

T S E T R C M Y N N F E A K H A O U G E

N A D I C N P I I M O N M I P L D E K M

E L D C Q S R A N V A H N F I A A V N

K A K I F I D Q R G O G I T V B R L I W

R I A W W O S T W B T Y E O W A O A I E

C I N S I N S Y K O D X A N P M L O A N

W D S M A I N E N S A E X D F A O S W L

G Z A V G L F D Q S S U Q Q J K C R A Z

U F S F K L J C A L G Q L R X D M R H Y

R Y D W X I F M M I S S I S S I P P I O

1 -ALABAMA
2 -ALASKA
3 -ARIZONA
4 -CALIFORNIA
5 -COLORADO
6 -FLORIDA
7 -GEORGIA
8 -HAWAII
9 -IDAHO
10 -ILLINOIS
11 -IOWA
12 -KANSAS
13 -KENTUCKY
14 -MAINE
15 -MISSISSIPPI
16 -MONTANA
17 -NEVADA
18 -NEW MEXICO
19 -OHIO
20 -OREGON
21 -TEXAS
22 -VIRGINIA
23 -WASHINGTON
24 -WISCONSIN
25 -WYOMING

TI—BASE - From INSCEBOT
TUTORIAL By Martin Smoley
NorthCoast 99'ers — July 25, 1988

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as its free.

The last article I wrote on TI-Base was a review in the July/Aug. newsletter. In that article I told of many problems I had with the PRINT command and other functions of TI-Base. I also said that I thought these problems would be corrected, and many improvements would be made. I'd like to say that the second of those two statements is now the most important. I received (Via Deanna Sheridan) a copy of TI-Base Version 1.02 and a four page letter from Dennis D. Faherty it's author. In the letter he related to 10 previous errors that had been corrected (one of which was the PRINT error) and to a multitude of improvements and refinements he wanted to make on TI-Base. This information has made me ecstatically happy. I feel that TI-Base will become as popular as TI-Artist and at some point will be so popular that you will be able to get COMMAND FILE routines from your club library just as you can now get Multiplan Screens or Extended Basic programs. TI-Base is a great enhancement to the 99/4A.

And now the TUTORIAL folks. First some housekeeping. The letters TIB will refer to TI-Base. MT: will signify the beginning of some text which should be considered Marty's Theory. Marty's Theory should not be taken as fact, but as my interpretation of an item. FYI: designates text that is For Your Information. FE will stand for For Example. DP will stand for Dot Prompt. <E> means press ENTER. (FEL) means Further Explanation Later, and last for now is ">", the greater than sign. I will use ">" when program segments are displayed at the left of every line. The position immediately to the right of the ">" will be column one. Take the example >12345. You should think of the number 1 as column one. The > does not exist. It is for reference only, the same as when you type in an XBasic program, at the head of each line you see > but it is not part of the program.

Let's get started. The first thing you do is make backups or copies of the original TIB disks and put the originals away in a safe place. If the originals arrived without the write protect slots on the disks being covered, do that first, then make your copies. The program will read and write to all of the disks used in the database process so you cannot writeprotect them. This means that you shouldn't use original disks and you should make copies of everything at the end of every work session. Backing up doesn't matter a lot at this point, but if you lose a data base with three or four hundred names in it, and you don't have another copy, you're in for some agonizing re-appraisal.

Having stashed the originals put your copy of the TIB system disk in Drive 1 and a newly initalized SS/SD disk in Drive 2. Then select Extended Basic and TI-Base will auto load. It takes a couple of minutes so be patient. After loading, TIB will ask for the date. This will be MM/DD/YY or Month, Day, Year. Enter the date, and use zeros, it's good procedure. FE, July 9, 1988 would be 07/09/88. TIB will then save the date and DO the program called SETUP. FYI: In this system DO replaces the XBasic RUN (more or less). When SETUP is executed you will be left with a bunch of junk on the screen and a dot " . " at the bottom left corner of the screen with the cursor flashing next to it (see SCREEN ONE). FYI: I will at least partially explain any new item we encounter as they occur. I will also try to proceed "Top-Down" in programming and explanation.

```
>001 *          Welcome to TI-BASE
>002 *    QUIT will terminate TI_BASE
>003 *
>004 SET DATSISK=DSK2.
>005 DISPLAY STATUS
>-------------------------------------
>DATDISK  = DSK2.       Database files on DSK2.
>PRGDISK  = DSK1.       TIB System Disk = DSK1.
>PRINTER  = PIO.        Printer port PIO/RS232 etc.
>LINE     = 080         Printer page width (Columns)
>PAGE     = 056         Printer page length (Lines)
>HEADING  = ON          Print all headings
>TALK     = ON          Echo commands to the screen
>SPACE    = 01          Space between fields
>RECNUM   = ON          Show record numbers
>LSPACE   = 0256        Space available for LOCALS
>DATE     = 07/09/88    This is the Date you Entered
>-------------------------------------
>006 *       FUNCTION (7) for help.
>007 RETURN
>.                [ SCREEN ONE ]
```

All of the lines with line numbers (001-007) are part of the command file called SETUP. The lines without numbers are part of the STATUS display. Lines 1, 2, 3, and 6 are comment lines and are made comment lines by placing an asterisk "*" in column one of any line. IMPORTANT: Line 2 could be misleading. QUIT does not refer to Fctn (Quit) in any form. You must never force the machine to quit or reset before you leave TIB by the proper procedure. Line 2 means type QUIT at the dot prompt and press enter. TIB will then take care of it's housekeeping (close all files, etc.) and exit to the TI system. Lines 4 and 5 are actual commands which can be included in a command file or typed in at the DP. FE type the following exactly at the DP

```
>SET DATDISK=DSK1. <E>
>CLEAR <E>
>DISPLAY STATUS <E>
```

You'll notice that the word CLEAR, cleared the screen and DISPLAY STATUS brought back the stuff between the dashed lines. You should also see that DATDISK now equals DSK1 (if it didn't work, type it in again and be careful for spaces etc. When you have made it that far type the following. This should reproduce the original SCREEN ONE.

```
>DO SETUP <E>
```

Continued Next Page.

The RETURN in line 7 returns the system to the level prior to this program section. You typed DO SETUP from the DP so when the RETURN is encountered we are returned to the DP. If we executed SETUP from another command file, when we hit the RETURN the program would have gone back to the file that called it (FEL). Let's do some housekeeping. Type in the following.

```
>COPY DSK1.SETUP/C DSK2.SETUP/C  <E>
```

When you see the message "ready devices, press ENTER", just press enter. The command you have just entered will then go to drive one and run a subprogram of TIB to preform the COPY function. That subprogram will then COPY the command file named SETUP/C from drive 1 to drive 2. The first DSKx designates "FROM" and the second DSKx designates "TO" a drive number. The first name "SETUP/C" is the complete name of the setup command file and must be used in this instance. You recall that when a DO SETUP/C from drive 1 to drive 2, the first DSKx designates "FROM" and the second DSKx designates "TO" a drive number. The command is issued (DO SETUP) the /C is not included in the name (FEL). The second name, or the name you're copying to, can be any name you wish (up to 10 letters)(FEL). FYI: We have copied setup to drive 2 because if you type DO SETUP at any time TIB will look for it there (try it and see). You should get a feel for what's on which disk as we go along. "OK, let's CREATE a database." Type in the command lines as you see them to the left.

```
>CLEAR <E>
>CREATE TNAMES <E>
```

When you type CREATE TNAMES and press enter, you will immediately see [SCREEN TWO].

arrows to move, enter to advance

FIELD	DESCRIPTOR	TYPE	WIDTH	DEC
1	_____	_	___	__

[SCREEN TWO]

This is the screen in which you tell TIB the size and shape of the database you would like it to create for you. This is actually called the STRUCTURE of the database, and that is why the command DISPLAY STRUCTURE will give you a screen like this one, but with all the pertenent information filled in. NOTE: A database must be in use at the time. The DESCRIPTOR is the name you will call a particular item, such as Last-Name, First-Name, Middle-Initial, etc. MT: If you can keep these names short, like LN for Last-Name, or MI for Middle-Initial, later on when you are using those names to preform different tasks you will not have as much typing, and you'll be able to get more on each line, plus (memory space is tight) (FEL). The TYPE is a one character entry, either N, C, or D. N stands for Numerical, C is Character, and D means Date. MT: Make all your fields C for Character unless you plan on performing a mathematical function on it. For example, the zipcode is all numbers but it should still be designated C for Character. The Date designation is used when you want the computer to enter a date for you, or when you are going to enter a date in the form MM/DD/YY. I do not want to go into this theory so early in the tutorial. Instead let's get going on TNAMES.

I have created a database call TNAMES using the information displayed in [SCREEN THREE]. Type in the data exactly as you see it so we can move along.

arrows to move, enter to advance

FIELD	DESCRIPTOR	TYPE	WIDTH	DEC
1	LN	C	15	
2	FN	C	15	
3	MI	C	2	
4	SA	C	25	
5	CT	C	20	
6	ST	C	2	
7	ZP	C	5	
8	PH	C	12	
9	XP	C	5	
10	GP	C	5	
11	ID	N	7	0

[SCREEN THREE]

When you are entering information these keys are active.

FCTN 1 = Del. Char.	Delete one character
FCTN 2 = Ins. Char.	Insert one character
FCTN 3 = Del. Line	Delete complete line
FCTN 4 = Ins. Line	Insert a complete line
FCTN 5 Not Used	
FCTN 6 Not Used	
FCTN 7 = AID	Brings up the help screens
FCTN 8 = Save/End	Saves the STRUCTURE
FCTN 9 = Escape	Discards the STRUCTURE
ENTER = Next Col.	Moves to the next column
Arrow Up Active	Moves to previous line
Arrow Left Active	Moves <= one Char./Column
Arrow Right Active	Moves => one character only
Arrow Down Active	Moves down one line

If you are apprehensive, type CREATE XP <E>. When the screen comes up type in all kinds of junk. Arrow up, down and backwards. When you see how it works press FCTN 9. All your garbage will be thrown away and you can start in on TNAMES. While you are entering the information for TNAMES as in screen three the only place there may be a question might be in field 11. When you get to the TYPE column, enter N and press enter. At that point the cursor will jump to the WIDTH column and the DEC or DECIMAL column will be highlighted. This only happens when you designate N for numbers. You then type 7 in the width column and when you press enter the cursor will jump to the DEC column. You now enter the number of decimal places you desire. If you were planning on dollars and cents, you might use 2 as the number of places. We are using a whole number so enter a 0 for no decimal places. When you have entered field 11 press FCTN 8 and TIB will create TNAMES for you and ask if you would like to enter some data at this time. If you answer N for no, you will be kicked back to the DP. If you have the stamina at this point, answer Y for yes and enter the data from my printout [SCREEN FOUR] at the top of page three of this tutorial. I have entered four ficticias names, and my own, in TNAMES. I will use this data in future tutorials.

Continued Next Page.

REC	LN	FN	MI	SA	CT	ST	ZP	PH	XP	GP	ID	ith
0001	Aardvark	Grant	E.	9995 State Rt. 84	Geneva	OH	44014	1-465-9876	02-88	NOCO	0717851	ith
0004	Smoley	Martin	A.	6149 Bryson Drive	Mentor	OH	44060	257-1661	02-89	NOCO	0713831	ith
0003	Jones	Quincy	W.	37285 Burgandy Laine	Mentor-on-the-Lake	OH	44060	257-1029	08-88	NOCO	0820871	ith
0002	Whitman	Raymond (Slim)	A.	2574 East 254th.	Eastlake	OH	44094	951-2345	09-88	NOCO	0921861	ith
0000	Vivannovitch	Flexxie	I.	111 E. 98th. St.	Cleveland	OH	91023	541-5415	05-88	NOCO	0712881	ith

[SREEN FOUR]

TI—BASE Tutorial Page 3

Having entered Y/es to enter data after the last screen, you should be in the APPEND mode, and you should see [SCREEN FIVE].

APPEND

```
                                        000
LN  _____
FN  _____
MI  _
SA  _____>
CT  _____
ST  __
ZP  _____
PH  _____
XP  _____
GP  _____
ID  _____
```

[SCREEN FIVE]

While entering data the previously described key functions are in effect. When you finish typing in the Last-Name (LN) pressing Enter will move you to the next field. You will notice that the numbers that run up at the far-right of each line are actually keeping track of your character position. The ">" at the end of line SA is telling you that there are more spaces for characters past the highlighted area. "In this case only one space." As you enter data and reach the end of the ID field, when you press Enter a new blank screen will come up. At that point the cursor will once again be in the first position to start entering another last name. If you are on the last data to be entered and at the end of the last field, do not press Enter. At that point you should press FCTN (8) to SAVE/QUIT. This does save, but it doesn't really quit, and you'll have to press FCTN(9) to get back to the DP. If you were worn out back when the question of entering data originally came up, you answered no and got out of the systen. You can now get back in by typing the lines to the left. The CLEAR is not really necessary in this case but helps me see any new screen messages without the extra clutter. NOTE: The EDIT is only usable when you already have data in the data base. I hope I have not been to confusing and you have been able to create the database and enter the data in screen four. If not, re-read this tutorial and consult your TIB manual. I'd like you to have a small database and be able to do something with it by the end of this tutorial.

```
>CLEAR <E>
>USE TNAMES <E>
>APPEND <E>
-------------------------
>CLEAR <E>
>USE TNAMES <E>
>EDIT <E>
```

Something I have not covered adequately up to this point is the phrase CLOSE ALL, and what's happening at the bottom of your screen in the highlighted area. I previously stressed the point that you must type the word QUIT at the DP in order to leave TIB. Doing so would cause TIB to look for and close any open databases before it quits to the TI system. When you are working with one database, and you would like to use another database you type CLOSE <E> at the DP. If you are working with several databases and wish to do something else, you type CLOSE ALL <E>. The highlighted area at the bottom of the screen will give you information on files that are open. This is particularly helpful when your screen is blank and the cursor is sitting at the DP. This information will consist of the name of a database which is currently open, and SELECTED (FEL), the record number which TIB is currently pointing at, and it will flash current system operations in the far right hand corner (FEL). My point is that if you see a name and some record numbers at the bottom of the screen, you should type CLOSE ALL <E), before starting any new major tasks. Assuming that you have managed to create the database named TNAMES and have typed in the information shown in screen four, I'd like to run through a couple things that should be enlightening. Type in the items at the left as usual. The system will give you messages as the data is being sorted, etc. Read the messages and observe the printout. I am attempting to show the unbelievable flexability of this program. Merely by typing in a few lines of text at the DP you can sort the data on a different field, and print out only the fields you want, in the order you want. At this point you probably get confused by the different nature of this programming language. When you have used it for a while you'll think it's the greatest record keeping system to come out for the TI, bar none. With the use of the APPEND mode you can add as many new records as you wish, and with the EDIT mode you can correct or change any information in the database. FYI: Before moving on I want to fill you in on SCREEN FOUR. In order to get that printout, I previously set my printer to condensed print. I then entered SET LINE=134 at the DP: 134 was the only length that worked properly (I tried several). Then I typed USE TNAMES <E> and PRINT ALL <E>. I don't know where the end characters in each line came from.

```
>CLEAR <E>
>CLOSE ALL <E>
>USE TNAMES <E>
>SORT ON FN <E>
>PRINT ALL FN,MI,LN
-------------------------
>SORT ON LN <E>
>PRINT ALL LN,FN,MI
-------------------------
>SORT ON ZP <E>
>PRINT ALL FN,MI,LN,ZP
-------------------------
>SORT OFF <E>
>PRINT ALL FN,MI,LN,ZP
-------------------------
>SORT ON XP <E>
>PRINT ALL FN,MI,LN,XP
>CLOSE ALL <E>
```

Continued Next Page.

Now it gets interesting. We are going to create a small program, or create a COMMAND FILE. However, create is not the right terminology. The phrase is MODIFY COMMAND (filename) <E>. Filename is any name you would like to call the command file. It should be eight characters or less in length, and do not add any of the identifiers you may have picked up along the way (/C). Just type everything to the left exactly as you see it. Take your time typing and allow time for the computer to do its job each time you press enter.

```
>CLEAR <E>
>CLOSE ALL <E>
>MODIFY COMMAND LBLS1 <E>
```

```
>* Command file LBLS1    "LABEL Prog."
>*
>SET TALK OFF
>SET RECNUM OFF
>SET HEADING OFF
>SET LINE=80
>CLEAR
>LOCAL TEMP C 40
>LOCAL BLNK C 1
>USE TNAMES
>SORT ON ZP
>TOP
> WHILE .NOT. (EOF)
>   REPLACE TEMP WITH "                ";
>      ! "  Exp. Date " ! XP
>      PRINT TEMP
>      PRINT BLNK
>   REPLACE TEMP WITH TRIM(FN) ! " ";
>             ! MI ! " " ! LN
>      PRINT TEMP
>      PRINT SA
>  REPLACE TEMP WITH TRIM(CT) ! ", ";
>             ! ST ! "   " ! ZP
>      PRINT TEMP
>      PRINT BLNK
>   MOVE
> ENDWHILE
>CLOSE ALL
>SET TALK ON
>SET RECNUM ON
>SET HEADING ON
>RETURN
```

```
>FCTN (8)          This will save the command file.
>DO LBLS1  <E>     This will run the file.
```

The information starting with CLEAR and ending with DO LBLS1 is everything you must type in to create and run a small program that will produce mailing labels from the database named TNAMES. It is that easy, and yet it is quite complicated. I will take the last half page of this article to give you some idea what's going on. The rest must wait until next month. I hope that what you have done so far has run successfully and your mind hasn't turned to mush.

The line MODIFY COMMAND LBLS1 <E> is the line that invokes TIB's Editor. This establishes that a command file is being created and will (if successful) be save to the DATDISK under the name LBLS1. At the time the file is saved the identifier /C will be attached to the name LBLS1 to produce LBLS1/C. This is why you cannot use 10 characters in the file name. Once you are in the editor the previously described keys are active (F1,F2,F3, Arrows, etc.). Lines that start with an asterisk "*" are comment lines. FYI: Don't use more than a couple comments, they eat up memory (FEL). All of the lines that SET something OFF are housekeeping. LOCAL TEMP C 40 initializes the variable named TEMP. TEMP will hold up to 40 characters (C). The variable BLNK can hold 1 character (C). At this point both variables are initalized blank or empty. We will refill and/or use them later. In the next three lines we are telling TIB to USE TNAMES and SORT that database ON the Zipcode field (ZP). When it is done we want it to go to the TOP, or beginning of the database. The next part of the program is a chunk. The chunk I refer to is everything from WHILE to ENDWHILE inclusive. This is the part of our program that does most of the work. When our program executes the word WHILE it does the whole line. This actually says to TIB, WHILE you do and ENDWHILE. If you do encounter the (EOF), or in this case the end of the database, then go to the next line after the ENDWHILE. The next line inside the loop will REPLACE the empty space in the variable TEMP with a bunch of blank spaces, the phrase " Exp. Date " and the club members Expiration Date (XP). The vertical lines "!" mean concatenate or stick together, the same as "&" in Extended Basic. So all three of those items are put into TEMP. Those items are then printed with the line PRINT TEMP. PRINT BLNK is the equivalent of "print a blank line". The next REPLACE takes FN (First Name), TRIMs off all the trailing blank spaces, sticks one space back (" "), attaches MI and another space (" "), puts LN (Last Name) on the end of that and sticks the whole mess into our variable TEMP. Now you see why TEMP had to hold up to 40 characters. The semicolon ";" at the end of these long lines is telling TIB that I couldn't get it all on 1 line and it should look for more on the next line down. TEMP is then printed as before. SA or Street Address is printed directly with no fancy stuff and the process is repeated for CT, ST and ZP. The blanks are thrown in for proper spacing to the next label. MOVE, moves the database to the next record and ENDWHILE sends you back to the WHILE statement to start over with the next name and address. The rest of the program is rather boring. When you finally run out of records the program jumps past all this to the CLOSE ALL. TNAMES is closed, everything you turned OFF is turned ON again, and the program is over. IMPORTANT, next month I will work with larger programs, using the FunnelWeb Editor/Assembler Editor. The program on this page (LBLS1) is about the best you can write using the Modify Command Editor. I will also get into the use of printer control codes. Control codes can be imbedded in the program with the FNLWB Editor, but not with the TIB Editor. I will cover some of the (FEL)s, Further Explanation Later and I will pr[illegible] everything many times. In TIB there are sever[illegible] write a program to accomplish the same task. [illegible] that situation I will compare the previous [illegible] should give you more contact with TIB logical procedures.

Continued Next Month.