West Jax 99er News

FEBRUARY 1989

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

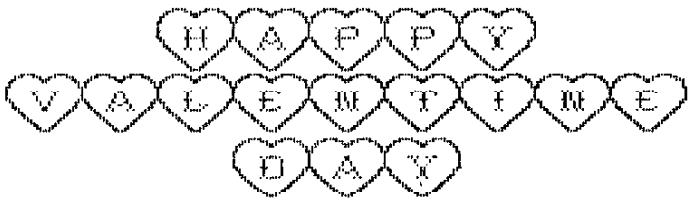
President...Rick Felzien.....(904) 772-9162
Treasurer...Thomas LeMay.....(904) 282-5220
Librarian.....Zach Ziegler.....(904) 389-2194

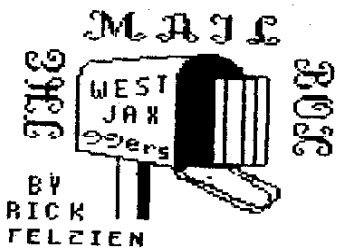
For newsletter suggestions and submissions, contact Rick Felzien.

I would first like to express my regrets that we did not have a newsletter last month, but as I said before, with an extremely small group it is hard to have material for a newsletter every month.

This month we have the usual mailbox article along with the Basic Assembler installment.

I have also included an article on the Star NX-1000 Rainbow, colour printer.





QB Monitor 99er Nov 88
1. TI-Writer tool box
2. Mod. Widget for superXB cart.

Spirit of 99 Jan 89
1. Ctting most from cassette II
2. Easy Grader program
3. TI-Writer part 14
4. Disk Fix
5. Maze maker program

Club 99 newsletter Nov 88
1. The mysterious sector zero

North Jersey 99ers Jan 89
1. Your INPUT-ing on me
2. XB * 12

Rocky Mountain 99ers Jan 89
1. Numeric representation
2. Assembly article

TI-D-BITS Dec 88
1. TI allophone speech
2. RGB and your TI
3. Imaginative programming

West Penn 99ers Dec 88
1. Tips for beginners
2. Disk drives *4
3. Putting Grom in the console

Eugene 99er TILT Dec 88
1. Review of MacFlix
2. Review of Triad
3. A word on Copyrights

SNUGletter Dec 88
1. Hard drive information

Ozark 99ers Dec 88
1. Review of 1000 words

Ozark 99ers Nov 88
1. Windows and Inverse Video

SFV 99ers Dec 88
1. Nice 3 column program

Central Pennsylvania 99ers Nov 88
1. Review of First Base
2. The Fairware decision
3. Electronic invasion

San Diego Comp. Soc. Nov 88
1. Squeezing Assembly

Erie 99ers Nov 88
1. Multiplan *4
2. Getting most from cassette

CIN-DAY news Nov 88
1. Basic Banners program
2. Tech Note on origional TI drive
3. How accurate is the TI

West Penn Newsletter Jan 89
1. patches for TI-Base V 2.0
2. 64k on the 16 bit bus
3. Myarc hard and Floppy controller
4. A lightpen project

Houston 99'er news Oct 88
1. Buying guide to modems

Houston 99'er news Nov 88
1. 3.5" disk drives

Houston 99'er news Dec 88
1. TI-Base part two

Southwest 99'ers Feb 89
1. Accuracy of TI sound chip

N.O.V.A. 99'er news
1. Desktop publishing part III
2. TI hacker's conversion chart

QB monitor Dec 88
1. Powering additional drives
2. Make your own power supply

Nutmeg 99'ers Jan 89
1. Extended Basic tools
2. The nonprogrammers guide

LITI users news Feb 89
1. Rave 99 memory enhancement sys.

TICO news Dec 88
1. Review of Form Shop

Cleveland Area news Jan 89
1. several TI-Base tutorials
2. Plus tutorial

Ottawa 99'er newsletter Jan 89
1. Expanding ExBasic's powers
2. Fast ExBasic

LA Topics Oct 88
1. TI-Base tutorial I
2. Cheat mode for TI-Runner
3. Beginning Forth

LA Topics Feb 89
1. TI-BAse tutorial II
2. Beginning Forth

CIN-DAY news Dec 88
1. TI-Writer tips I

CIN-DAY news Jan 88
1. TI-Writer tips II
2. Key return code chart
3. TI-Keys review

Front Ranger Jan 89
1. FirstBase review

The STAR NX-1000 Rainbow Color printer

By
Rick Felzien

I was very pleasantly surprised recently when my wife let my procure a new printer. It just happened that the Navy Exchange at NAS Jax got a shipment ot the Rainbow printers and placed them on sale at a very reasonable price.

I was like a kid with a new toy when I read the book and found that it had all the nice fonts plus color. It will also do, in addition to the expanded print, double-height, double-sized, and even Quad-sized text.

It sort of scared me when the book said that there was no longer the tradidional STAR mode. It is designed to emulate the Epson Graftrax or the IBM Proprinter. The download character ram is used as a print buffer unless one of the DIP switches is set in another position. The print buffer has furned out to be a blessing in disguise. I had feared that the IBM character and drawing set would be a real bear to access, but I found even this to be a breeze with a little practice. In fact while playing around with translating the demonstration program from IBM basic, I learned a couple tricks about printer codes in ExBasic that I hadn't known.

The first thing you notice when unpacking the Rainbow is that there is no tractor feed on top (panicksville), but on checking the book you find that it is below the platten (roller). The paper feeds through the rear and to the tractor and then to the platten. This saves wasting a sheet of paper every time you print something. There are also codes for reverse micro-feeds as well as reverse linefeeds, which could lead to some interesting printing techniques.

Not only will the NX-1000 Rainbow give you pica, elite, condensed pica, condensed elite but also you have the following NLQ fonts: courier, sanserif, orator with small caps, orator with lower case, but all can be done in italics as well as condensed. It will also do proportional and centered text printing. Most of these can be set with the switches near the on/off line switch.

Another unique feature is paper-parking which retracts the fan-fold paper almost to the tractor head to allow for single sheet printing and then when you are finished with single sheets, you can run the fan-fold back into place. It has a handy paper guide built in for feeding in single sheets.

I have enclosed a sheet from the demo program, too bad I don't have access to color pronting to show how nice the colors are. Now if only the graphics and graphing software for the TI only had provisions for color printers. Never fear, I had to call Dennis Faherty concerning his TI-Base program and while I was at it I talked to his son Chris. I had mentioned that it sure would be nice to do color with TI-Artist, and he said he is working on a new version of Artist which, among other nice new features, will be capable of addressing color printers. Now isn't that great, I guess all you have to do is ask and your dreams can come true. After I talked to Chris Faherty I decided to write to Mike McCann and inquire into the possibility of getting color capability in his fine Business Graphs 99 package but have not gotten an answer yet. Just think how nice it would be to be able to do color graphs and charts.

Anyway, back to the printer. Depending on DIP switch configuration, you can use download RAM and use IBM mode, and all seem to be fairly accessable by the TI programming.

As far as color text, it is accomplished just as with any printer capability with .TL coding and it works great. I have even tried creating a form with TI-writer using .TL and the IBM character set and even this works flawlessly. Like I said, I am like a kid with a new toy, and am I having fun!

This is my third Star Micronics product. I first had the old standby Gemini 10X, then an SG-10, and now the NX-1000 and have never had a complaint about the operation and they have all proven to be extremely reliable. The only problem I have ever had is that the print head went on the Gemini after four years of personal printing as well as a newsleter almost every month. Not bad huh!

Incidentally when I got the NX-1000 I found out how easy it is to make your own PIO printer cable. I am working on an article about it for the next newsletter.

Another thing that I must mention is that I almost got the demo program translated. I am having trouble with the dot graphics routine and would appreciate hering from anyone who may have licked it.

The address is:

Rick Felzien
6872 Peter Pan Pl.
Jacksonville, Fl
32210

Type styles are:
Draft characters,
Courier characters,
Sanserif characters,
ORATOR WITH SMALL CAPITOLS, OR
with lower case characters,

and ITALICS for all styles.

Print pitches are:
Pica pitch. Elite pitch,
Condensed pica pitch, Condensed elite pitch,
Proportional spacing for all pitches.

Expanded, Double-height,
Double-sized.

Quad-sized.

Colour printing:
RED, BLUE, VIOLET,
ORANGE, GREEN, and BLACK.

Various line and character spacings:

THE SPACING HAS CHANGED
THE SPACING HAS CHANGED
THE SPACING HAS CHANGED

THE SPACING HAS CHANGED
THE SPACING HAS CHANGED
THE SPACING HAS CHANGED
THE SPACING HAS CHANGED
THE SPACING HAS CHANGED

Other features:
Emphasized, Double-strike,
Underlining, Overlining,
Superscript, Subscript.

T H E B A S I C A S S E M B L E R #7 By Steve Peacock

USING A CHARACTER IN A MAZE (READ THE SCREEN DO NOT GO THROUGH THE WALL)

This month we will use several of the things that have been taught in past months. With this program, a maze is created and you move your 'man' through it. The main new thing is how to read the screen and not move if you are up against a wall. The BASIC command 'CALL GCHAR' is used to read a character, on the screen, and then print the animated character, if the new position is valid. The program will not permit the printing, if the new position in not valid. In the assembly version the command 'VSBR' is used. In using this routine, the following must be kept in mind. First store the current row and column position in a holding variable. Then update the row or column. Next read the row and column to see if you can print there. If you can, then print. If not retrieve the stored row and column.

When typing in the assembly version of the maze, please note that most lines start with two spaces and also end with two spaces. These spaces must be included in order to make the total number of bytes correct.

In line 440, 470 and 480 of the BASIC version there is a variable 'UCR', this is the UnChangedRow. Likewise in line 440, 490 and 500 the 'UCC' is the UnChangedColumn.

Please save this program. I plan to add to it in the future. If you type it in exactly as it is printed you will be able to make the changes without any trouble.

```
*****************************************************************************
*
*PROGRAM BA7A==>Basic Assembler #7 Assembly Version
*USING A CHARACTER IN A MAZE (READ THE SCREEN DO NOT GO THROUGH THE WALL)
*(C)1985 S. PEACOCK   (Save this program, we will add to it later.)
*
*****************************************************************************
       REF  VSBW,VSBR,KSCAN,VMBW,VMBR,VWTR
       DEF  START
START  LI   R0,>0958      *REDEFINE CHARACTER '+' (>0958 ADDRESS)
       LI   R1,DF1        *DF1 IS THE HEX CODE FOR THE BARRIERS
       LI   R2,8          *8 BYTES TO WRITE
       BLWP @VMBW
DF1    DATA >CCCC,>3333,>CCCC,>3333
       LI   R0,>0706      *WRITE TO REG. 7, CHANGE SCREEN TO DARK RED
       BLWP @VWTR
       LI   R0,>0385      *CHANGE COLOR OF SET 2
       LI   R1,>AF00      *DARK YELLOW ON GRAY
       BLWP @VSBW
       LI   R0,>0384      *CHANGE COLOR OF SET 1
       LI   R1,>6600      *DARK RED ON DARK RED (FOR THE SPACE/CHAR 32d 20h)
       BLWP @VSBW
*************************PRINT MAZE. MAZE CAN BE CHANGED BY ARRANGING THE
       LI   R0,0          *'+' TO DIFFERENT POSITIONS. WHEN PRINTING THE
       LI   R1,MAZE       *FIRST 'X', ADJUST POSITION IF NEEDED.
       LI   R2,672
       BLWP @VMBW
```

```
**********              **COLUMNS !**  NOTE: SPACES IN COLUMNS 1,2,31,32
**********              **        V**        MUST BE INCLUDED
**********            1         2         3              TOTAL
**********  12345678901234567890123456789012             BYTES
MAZE    TEXT  '   ++++++++++++++++++++++++++++++++   ' * 1      32
        TEXT  '   +                              +   ' * 2      64
        TEXT  '   + ++++++++ + + + + ++++++++ +   ' * 3      96
        TEXT  '   + +       + + + + + +      + +   ' * 4      128
        TEXT  '   +   ++++ + + + + + + + +++    +   ' * 5      160
        TEXT  '   + +       + + + + + + +      + +   ' * 6      192
        TEXT  '   + + ++ + + + + + + + + + + + +   ' * 7 R    224
        TEXT  '   + +       + + + + + + +      + +   ' * 8 O    256
        TEXT  '   + + ++++   + + + + +   +++ + +   ' * 9 W    288
        TEXT  '   + +       + +     + +     + +++'  *10 S    320
        TEXT  '+++ + ++ +++ + + + + + +++++ +     '  *11<-   352
        TEXT  '     +           + +           + +++'  *12     384
        TEXT  '+++ ++++++++++ + + +++++++++ +   '  *13     416
        TEXT  '   +                 + +           +   '  *14     440
        TEXT  '   + ++++++++++ + + +++++++++ +   '  *15     480
        TEXT  '   +   +     + + + + + + +   +   +   '  *16     512
        TEXT  '   + +   ++   + + + + +   +   + +   '  *17     544
        TEXT  '   + + +     +   + + +   + + + +   '  *18     576
        TEXT  '   + ++++++++++ + + +++++++++ +   '  *19     608
        TEXT  '   +                              +   '  *20     640
        TEXT  '   ++++++++++++++++++++++++++++++++   '  *21     672
        LI    R0,304          *PRINT FIRST 'X'
        LI    R1,>5800        *
        BLWP  @VSBW           *
        CLR   R1
LP      LI    R1,>0100        *>0100 TO READ JOYSTICK #1
        MOVB  R1,@>8374       *TO SET UP KSCAN TO READ JOYSTICK
        BLWP  @KSCAN          *SCAN THE KEYBOARD (JOYSTICK)
        CLR   R1              *CLEARS REG. 1 WHERE TO VALUE @>8376 WILL BE PLACED.
**************************RIGHT BYTE MUST BE ZERO FOR COMPARISON.
        MOVB  @>8376,R1       *CHECK Y RETURN
        CI    R1,>0400        *Y RETURN OF UP POSITION (04)
        JNE   T1              *JUMP IF NOT EQUAL TO T1 (NEXT CHECK)
**************************IF EQUAL PUT THE UNCHANGED POSITION IN REG. 8, THIS
**************************MAY BE NEEDED LATER, IF THE 'X' IS NOT TO BE MOVED
        MOV   R0,R8           *REG. 8 WILL HOLD THE UNCHANGED POSITION
        AI    R0,-32          *CHANGE REG. 0, DECREASE BY ONE ROW (MOVE UP)
        JMP   PG              *JUMP TO PRINT ROUTINE
T1      CI    R1,>FC00        *Y RETURN OF DOWN POSITION (-04d FCh)
        JNE   T2              *SAME AS ABOVE
        MOV   R0,R8
        AI    R0,32
        JMP   PG
T2      MOVB  @>8377,R1       *CHECK X RETURN
        CI    R1,>0400        *X RETURN OF RIGHT POSITION (04)
        JNE   T3              *SAME AS ABOVE
        MOV   R0,R8
        INC   R0
        JMP   PG
T3      CI    R1,>FC00        *X RETURN OF LEFT POSITION (-04d FCh)
        JNE   LP              *SAVE AS ABOVE
        MOV   R0,R8
        DEC   R0
```

```
**************************START OF PRINT ROUTEIN
PG      MOV   R0,R9           *REG. 9 HOLDS THE CHANGED POSITION
        CLR   R1
        BLWP  @VSBP           *READ TO SEE IF REG. 0
        CI    R1,>2B00        *HOLDS THE '+' IF IT DOES THEN WE CAN NOT PRINT
        JEQ   CB              *THE 'X' IN THE WALL (CB-CHANGE BACK)
        CLR   R1              *IF NOT CLEAR REG. 1
        MOV   R8,R0           *MOVE REG. 8 (THE UNCHANGED POSITION) TO REG. 0
        LI    R1,>2000        *PRINT A BLANK ON CURRENT POSITION
        BLWP  @VSBW
        CLR   R1              *CLEAR REG. 1
        MOV   R9,R0           *MOVE REG. 9 (THE NEW POSITION) TO REG. 0
        LI    R1,>5800        *PRINT A NEW 'X'
        BLWP  @VSBW
        LI    R4,6000         *
        DEC   R4              *A DELAY LOOP OF 6000d *TRY WITHOUT THIS LOOP
        JNE   $-2             *                      *FUN SPEED!!!
        JMP   LP
CB      MOV   R8,R0           *CHANGE PRINT POSITION BACK IF WALL IS HIT
        JMP   LP              *JUMP BACK TO MAIN LOOP
        END
```

```
100 REM PROGRAM BA7B==>Basic Assembler #7 Basic Version
110 REM USING A CHARACTER IN A MAZE (READ THE SCREEN DO NOT GO THROUGH THE WALL)
120 REM (C)1985 S. PEACOCK (Save this program, we will add to it later.)
130 REM YOU MAY WANT A 'CALL CLEAR' HERE
140 READ DF1$
150 CALL CHAR(43,DF1$)
160 DATA CCCC3333CCCC3333
170 CALL SCREEN(7)
180 CALL COLOR(2,11,15)
190 DISPLAY AT(1,1):"++++++++++++++++++++++++++++++++"
200 DISPLAY AT(2,1):"+                              +"
210 DISPLAY AT(3,1):"+ ++++++++ + + + + +++++++ +"
220 DISPLAY AT(4,1):"+ +       + + + + + +      + +"
230 DISPLAY AT(5,1):"+   ++++ + + + + + + +++    +"
240 DISPLAY AT(6,1):"+ +       + + + + + +      + +"
250 DISPLAY AT(7,1):"+ + ++ + + + + + + + + + + +"
260 DISPLAY AT(8,1):"+ +       + + + + + +      + +"
270 DISPLAY AT(9,1):"+ + ++++   + + + +   +++ + +"
280 DISPLAY AT(10,1):"+ +       + +     + +     + +"
290 CALL HCHAR(10,31,43,2)
300 CALL HCHAR(11,1,43,2)
310 DISPLAY AT(11,1):"+ + ++ +++ + + + + +++++ +"
320 DISPLAY AT(12,1):"  +         + +         + +"
330 CALL HCHAR(12,31,43,2)
340 CALL HCHAR(13,1,43,2)
350 DISPLAY AT(13,1):"+ ++++++++++ + + +++++++++ +"
360 DISPLAY AT(14,1):"+           + +           +"
370 DISPLAY AT(15,1):"+ ++++++++++ + + +++++++++ +"
380 DISPLAY AT(16,1):"+   +     + + + + + +   +   +"
390 DISPLAY AT(17,1):"+ +   ++   + + + +   +   + +"
400 DISPLAY AT(18,1):"+ + +     +   + +   +   + + +"
410 DISPLAY AT(19,1):"+ ++++++++++ + + +++++++++ +"
420 DISPLAY AT(20,1):"+                          +"
430 DISPLAY AT(21,1):"++++++++++++++++++++++++++++++++"
440 R,UCR=10 :: C,UCC=17
450 CALL HCHAR(R,C,88)
460 CALL JOYST(1,X,Y)
470 IF Y=4 THEN UCR=R :: R=R-1 :: CALL GCHAR(R,C,A):: IF A<>43 THEN GOTO 520 ELS
E R=UCR
480 IF Y=-4 THEN UCR=R :: R=R+1 :: CALL GCHAR(R,C,A):: IF A<>43 THEN GOTO 520 EL
SE R=UCR
490 IF X=4 THEN UCC=C :: C=C+1 :: GOSUB 560 :: CALL GCHAR(R,C,A):: IF A<>43 THEN
 GOTO 520 ELSE C=UCC
500 IF X=-4 THEN UCC=C :: C=C-1 :: GOSUB 580 :: CALL GCHAR(R,C,A):: IF A<>43 THE
N GOTO 520 ELSE C=UCC
510 GOTO 460
520 CALL HCHAR(UCR,UCC,32)
530 CALL HCHAR(R,C,88)
540 IF R=12 THEN R=11
550 RETURN
560 IF C=1 THEN R=11
570 RETURN
580 END
```