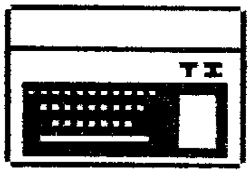
# TI CLUB OF OXNARD NEWSLETTER

VOLUME 5 NUMBER 6

Editor: Jeff Asenas



If this newsletter is late, It is due to a 10 page essay I have been working on since the last meeting. My apologies to all, I will try to be more organized next month as there will be no more school work!

A couple months ago (December '88 to be exact), I noticed an ad in MICROpendium from T.A.P.E. (1439 Solano Pl. Ontario CA 91764, (714)989-9906) that said they were having a super close-out sale on their Mechatronics items and this ment low, low fire sale prices. The item that caught my eye though was the TI Mouse. I have always envied the easy use of a mouse when I go to my friends houses who have Amiga's and MAC's, etc. and thought wouldn't it be nice to have one on the TI??? So I rushed the mag. to my dad (he has all the money! heh, heh!) and he said lets go for it. So just to be sure of getting the item, we (he, I mean) ordered it C.O.D.

After months and months of waiting, plus talking to Franz at the Fest-West back in February, the mouse arrived in a big box a couple days ago. The total came to a little over $60 including the C.O.D. fee's.

Almost as if I had gone through this scene a thousand times, I rushed into the kitchen, grabed a knife out of the silverware drawer and cut the packing tape. Inside was a 20 page manual with a light cardboard cover. Forget the manual though, I wanted to see how the mouse worked.

I dug deeper into the styrofoam packing and came up with an adapter attached to a black rectangle. This is the power supply for the mouse. I also came up with a disk that said TI MOUSE on it and finally the mouse itself.

The mouse is sort of a rectangular object rounded off at the bottom to fit the contours of your hand. It has two grey buttons on the top (one is the mouse key and the other is the HOME key for future applications). On the bottom is a circular hole with the end of a grey rubber-like ball. This seems to be a quality job of designing!

On the disk is an assortment of programs. The first called TI-DOS and is a desktop environment with icons. Here you can run XB programs, view any data or text file and delete programs. All of this is done by pointing the mouse pointer at the file and then moving to another icon. A HELP file is also accessable from the screen.

The second file is a Calculator program with windows. It works exactly as a handheld calculator only instead of using fingers to push buttons you use the mouse pointer.

The third program is a Breakout game written in assembly. It runs with the mouse only and the object is to knock out all the blocks with a ball that you hit with a mouse controlled paddle. It is very good except the ball doesn't hit very sharp angles.

Also on the disk are the assembly support routines for writing an Extended BASIC program for use with the mouse (source code is included for assembly programers).

And finally is the external DSR-input file for use with TI-Artist. This works O.K. I say this because it doesn't work well in the enhancement mode and also the LINE mode of TI-Artist doesn't go to swift with the mouse. The mouse DSR that comes on the Artist Extras doesn't even work at all. It just crashes the program. Anyway, I will try and poke around the source for the mouse routine and see what I can come up with.

I will be showing this at the meeting, so be there!

# - WISDOM LINES -
# -BY CHARLES HILLEY

## -MULTIPLAN TUTORIAL-

The purpose of this column is to provide useful information in utilizing the MICROSOFT MULTIPLAN package. I will be refering to the Multiplan manual, throughout this column.

I. SYSTEM REQUIREMENTS:
- A. Bare Bones
  - 1. TI-99/4A Computer
  - 2a. Peripheral Expansion System
  - 2b. CorComp Micro Expansion System
  - 3. 32K memory expansion card
  - 4. One disk drive
  - 5. RS-232 card
  - 6. Printer.
- B. "Would Be Nice"
  - 1. Two or more disk drives (Add to basic set-up listed in A.)
  - 2. Ramdisk

II. HEADACHE PREVENTION:
- A. Back-ups
  - 1. Please be sure that you make copies of the original Multiplan program disk just in case of system or operator trouble. Please refer to your disk manager operations manual.

III. GETTING STARTED: (pg. 12)
- A. First power up system by engaging the peripherals, monitor, and the console.
- B. Insert the Multiplan cartridge into the slot found on the console, and press the corresponding number displayed on the monitor (depending on system layout).
  - 1. Screen colors can be adjusted accordingly by pressing the space bar.
- C. Place the backup disk in drive 1 and press "Enter". The computer then will display, on the monitor, numbers along the left side of the screen and across the top. At the bottom, the word "COMMAND" appears along with a list of choices (More on this in Chapter 1).
- D. Filenames:
  - 1. Maximum of eight(8) characters in length.
- E. IMPORTANT!!!
  - 1. Single drive users:
    - a. Leave program disk in drive except for the following commands: Print File, Transfer Load, Transfer Save, Transfer Delete, Transfer Rename, and eXternal Copy. These commands require the data disk to be in the disk drive.
  - 2. Multi-drive users:
    - a. Select Transfer Options command by pressing "T" and then "O". Advance to the setup field by pressing TAB (CTRL A). Choose the data file drive and type DSK2 (or 3) and press "Enter". I've found out that this procedure must be performed each time when the program disk is called on initial startup.
- F. Printers:
  - 1. Select Print Options command by selecting "P" and then "O". Press TAB (CTRL A) to advance to the setup field.
    - a. Defaults:
      - 1. Serial: RS232.BA=300
      - 2. Parallel: PIO
    - b. To change the serial configuration, type "RS232.BA=(insert baud rate here)". Please refer to page 14 in the Multiplan manual.

IV. Chapter 1: Fundamentals (pg. 17)
- A. Spreadsheet capacity:
  - 1. 255 rows long
  - 2. 63 columns wide
- B. The Direction Keys
  - 1. UP (FCTN E)
  - 2. DOWN (FCTN X)
  - 3. LEFT (FCTN S)
  - 4. RIGHT (FCTN D)

CONTINUED ON PAGE 6

## ASSEMBLY LANGUAGE
*******
by Adrian Robinson

Just a short one this time but one which can be quite useful for all you Assembly programmers and even for those Extended Basic types.

We are all familiar with those programs that will run only in drive #1. They do so generally because they need to access other files on the same disk and must, therefore, know the number of the drive in which they were loaded. A few of our favorites, such as BOOT and the latest versions of FUNNELWEB, are "boot drive tracking" programs, which means that they can be run from any drive and can determine for themselves in which drive they were loaded. Well, boot tracking is so very easy to accomplish that it is really surprising that so few programs do it. The only instructions necessary are :

```
        LI    RØ,>3EEB       (or >3FF5)
        BLWP  @VSBR
```

and the high byte of R1 now contains the drive number.

Whenever we access a disk file, and that includes running a program, our disk operating system (DOS) places the drive number in the byte at >3EEB in VDP RAM. It also places the drive number at >3FF5 followed by the file name in the next ten bytes, so we can track the file name as well as drive number.

Now, if any of you Extended Basic types have read this far, you too can have the advantages of boot tracking in your own programs. I have included an assembly utility to be used with XBasic programs. Just assemble and save this program. Then load the assembly program before running your XB programs, and issuing a CALL LINK("TRACK",D$) from your XBasic program will return the drive number and file name in D$. Just remember that a CALL INIT will wipe out the assembly routine from low memory.

A far better way is to save the assembly utility in XBasic format with one of the assembly saver template programs. Then you can merge your XB programs to the saved assembly program, the boot tracker program will be loaded with the XBasic and you will not have to worry about losing it.

If you happen to have a CorComp disk controller card, you can use its built-in VPEEK utility for boot tracking from XBasic or console basic without the need for the assembly program given here. I have included an XBasic program for the CorComp controller to illustrate this.

So, whether you program in Assembly or XBasic, let's get tracking.

```
* * * * * * * * * * * * * * * * * * * * * * * * *

*         BOOT TRACKING           *
*        Adrian Robinson          *
*           May  1989             *
* XBASIC CALL LINK("TRACK",A$)    *
***********************************
       DEF   TRACK
STRASG EQU   >2Ø1Ø
VMBR   EQU   >2Ø2C
MYWS   BSS   32
H3Ø    BYTE  >3Ø
       BYTE  11              String Length
DSK    BSS   11         String Save Buffer
       EVEN
TRACK  LWPI  MYWS
       LI    RØ,>3FF5  Address of Drive #
       LI    R1,DSK
       LI    R2,11
       BLWP  @VMBR   Drive # and Filename
       AB    @H3Ø,@DSK   Drive # to ASCII
       CLR   RØ             Not an Array
       LI    R1,1        First Parameter
       LI    R2,DSK-1        Length Byte
       BLWP  @STRASG         Assign to A$
       LWPI  >83EØ                GPLWS
       CLR   @>837C         Clear STATUS
       B     @>7Ø      Return to XBasic
       END

******************************************
```

```
10 CALL CLEAR :: CALL SCREEN(
5):: FOR I=Ø TO 12 :: CALL CO
LOR(I,16,1):: NEXT I
2Ø CALL INIT :: DELETE "LD-CM
DS" :: A$=RPT$(" ",1Ø)
3Ø CALL LINK("VPEEK")(16373,Ø
,N,A$)
4Ø A$="DSK"&STR$(N)&"."&A$
5Ø DISPLAY AT(8,5):"My name i
s now": :TAB(5);A$: :TAB(
5);"Rename me and run me": :
:TAB(5);"in another drive."
6Ø CALL KEY(Ø,K,S):: IF S=Ø T
HEN 6Ø
```

******************************************

UGOC

BULLETIN

BOARD

3ØØ/12ØØ/24ØØ BAUD

(714) 751-4332

## NOT QUITE BASIC

by N.Armstrong

I was going to follow last month's offering with a program leaving the blank record in, but not showing it when the list was scrolled. That program isn't finished. It's based on a linked list program by S.T.Holl from the old 99er magazine. (Whatever happened to them, anyway?)

Most of Professor Holl's programs were called "Pocket ______" because of their brevity. And that got me thinking about "Tinygrams", those XBasic programs that can be listed on one screen. So, I put one together (listed below). The REM statements showing title and author do not count as part of the screen.

This program is just a "conveyancesfffor transferring text files to the printer. It prints a two-column page in the same way the ROM is printed. The program proper comprises the second half of the Tinygram; the first half is a brief description of file preparation.

```
5 ! ::::::::::::::::::::::::
    :: TWO COLUMN PRINTER ::
    ::      TINYGRAM      ::
    ::   BY N.ARMSTRONG   ::
    ::::::::::::::::::::::::
10 CALL CLEAR :: PRINT "TINY
GRAM prnts 2-col, 80-linpage
 in 1-pass. Prep copy inTIW,
 LM0 40Cx160L. Keep linsi-4,
77-84,157-160 blnk, xcpt"
12 PRINT "for CTRL U prntr c
mds, ASCII27 66 2 27 77 6 27
 68 44 0  27 85 1 27 65 10 a
s example.Use PF to disk to
save file.Want to CONTINUE?"
14 ACCEPT AT(23,20):Z$
16 DIM PT$(160):: DISPLAY AT
(3,1)ERASE ALL:"PUT SOURCE D
ISK IN DRIVE 1": :"TYPE FILE
 NAME: P4" :: ACCEPT AT(5,17
)SIZE(-10):D1$ :: OPEN #1:"D
SK1."&D1$,INPUT :: FOR I=1 T
O 160 :: LINPUT #1:PT$(I)
18 NEXT I :: CLOSE #1 :: OPE
N #2:"PIO",VARIABLE 132 :: F
OR I=1 TO 80 :: C$=PT$(I)&CH
R$(9)&PT$(I+80):: PRINT #2:C
$ :: NEXT I :: CLOSE #2
```

Proper preparation of the text file is the key ingredient in this program. This takes three steps: One, data entry and formatting for width.; Two, formatting for length; Three, inserting print control codes. But before we discuss these, let's talk about page layout.

We use Elite type (12 characters/inch) and 80 lines/page. Of course we don't have print on all 80 lines. We want some white space at the top and bottom of the page. Also, if we use blank lines for paragraph breaks, our page doesn't look like a solid wall of type.

The 12 cpi type allows 96 characters per line on an 8-1/2-inch wide page, leaving 1/4-inch margins on each edge. Since we are using columns 40 characters wide, we have a 16-character space to divide between the gutter and margins.

The binding edge should be allowed more space than the outside edge of the page. Allowing 8 spaces for the binding edge, 4 for the gutter, and 4 for the outside edge makes a nice looking printed page. How do we get this layout? Come along.

Formatting the text 40 characters wide can be done in the Editor with CTRL R or CTRL 2 (use L at 0, R at 40) and word wrap. Hyphenate to fill long spaces at line ends. Process the file in the Formatter for right-hand justification (use FI;AD;LM 0;RM 39). Print the file to Disk.

Next, load the file back into Editor. Each line will have a Line Feed symbol. We don't need that so eliminate it with Replace String. There are other ways to get rid of that LF, but even if you already know how to use RS and CTRL U codes, this is good practice.

Now, start adding and deleting blank lines so that we end up with 160 lines for our printed page; lines 1 to 80 comprise the first column and lines 81 to 160, the second column. Leave lines 1-4, 77-84, and 157-160 blank. This leaves a 4-line top and bottom margin.

When we have the lines all spaced so they will occupy the correct position, we tell the printer what to do. We do this with control codes in the blank lines and embedded in the text. (The codes are not printed, so the lines are blank on the finished page.)

Control codes are made from combinations of the ASCII numbers 0 through 127; the first 32, often called the non-printables, are the Special Characters referenced on page 98 and shown on page 146 of the TI Writer Manual. These Special Characters are accessed through the CTRL U mode.

(Because the Special Characters do not print, we will refer to them by their name and/or ASCII number.) On the first line (line 1), we insert commands that establish the page layout: 27@ (RESET) 27B2 (Elite) 27M8 (LH Margin) 27D40 (TAB 52) 27G (Double Print) 27A10 (Line Space 10/72) 27U1 (Print one way) etc. These codes are for Gemini 10X. *END LINE 154*

## OBERON INTERNATIONAL OMNI-READER
By Art Byers, Cent. Westch. 99'ers

FROM HARD COPY PRINTED TEXT INTO YOUR 99/4A AS ASCII TEXT.
or
WHEN IS A BARGAIN A BARGAIN?

All of us have seen mail order ads for low cost TEXT scanners. These are not the kind of scanner that produces what is really a dot graphics, pixel by pixel, picture and will copy photo's, drawings, etc., but rather devices that scan one line at a time of printed or typed text and convert it to ASCII text, sending it to your computer via the RS232 port. They started out costing more than $1000 and ended up at well below $200 as a mail order closeout. Boy! Doesn't that sound like a bargain??!!

While visiting a friend, on a rainy day, at his business office, I stumbled upon one of these devices when I went to hang up my wet raincoat in his closet. It was tucked into a back corner and the box was coated with dust.

I'll cut the story a bit shorter by saying that I asked to borrow it to try out on my 99/4A. My friend was a bit rueful about the OMNI-READER. It had been purchased for his IBM Clone, mail order, for about $150. His words and attitude were, in essence, "Have fun with it. You'll find it rather primative".

Despite his disparagment, the scanner appeared to be a new TOY I could play with and since it was borrowed, play for FREE. Here is my evaluation:

The OMNI-READER consists of a large easel platform to hold the text to be scanned. It has a guide ruler running on a metal bar to enable it to slide from top to bottom, a scanning device that moves along the guide ruler, power supply, RS232C port and a series of dip switches to set baud rate etc. It also came with two EXCELLENT manuals. Reading the manuals gives you ALL the information you need. That's on the plus side.

On the other side, well it has lots of faults.

First it is very slow, much too slow!. It stops at the end of each line to "digest" the information before transmitting it to your computer via the RS232. An average "fair" typist could win a copying race vs the OMNI-READER, with ease.

Second, the scanner is moved manually - yes, literally by your hand motion. It will not work well if you go too fast or too slow. There is "play" in the way the scanner attaches to the guide ruler. This means it is almost impossible to scan a line the in the identical way twice in a row. This results in a "bad read" at times when you feel certain you have done everything correctly. What is just as frustrating, sometimes it works and reads very well and you are not sure exactly what you did right!.

Third, The placement of the guide ruler is CRITICAL. You just can't slide it down the page quickly. It must be positioned VERY carefully over each line of type. This slows down the process, which is already too slow. OUCH!!!!

Fourth, it will only read two type faces - Courier and Prestige, though it will read them in 10 and 12 pitch and it will compensate for oversize or undersize type.

The OMNI-READER was able to read the NLQ Roman type off my Epson LX800 but it often read the upper case "S" as a "5". Roman is apparently quite close in design to Courier. It came with a disk of software for an IBM compatable that included several other type fonts that could be loaded into the O/R using its "Learn" feature. Obviously, that disk would not help me with the 99/4A. If the device had worked very much better and easier than it did, I would have considered writing to England for the protocol on how to construct a type face and make up additional popular fonts. The manual does contain transparent overlays to help you recognize type fonts. -BUT - as my friend said, at the very start, it is too primitive and too much of a headache to use.

A day after I obtained the text scanner, Al Trudeau and I sat down and managed to get it up and running on my 99/4A without undue difficulty. We tried three methods of reading the output: TI-Writer, a popular terminal emulator program (which in this case was FAST-TERM- though apparently any one would have done from the TE-II module to Mass-Transfer or TELCO) and last I wrote a simple XBasic program that was able to accept the text from the reader.

Our initial step was to take my trusty old breakout box and configure a cable so that my computer would recognize the reader. The OMNI-READER manual gave a clear diagram of how to do this. We simply followed the manual.

TI-Writer was a failure, even though the box advertised that you could use the scanner with a word processor. To attempt load the data from O/R, we used the LF command and typed in RS232.BA=1200.DA=8.PA=N. This simply did not work. Writer looks for a file header, specifically DF/80 or DV/80, and there is no other way to load data into TI-WRITER other than LF and the keyboard. The reader did not have any programable cabability so that we could have simulated such a TI file header.

Terminal emulators, however, worked very well once they were configured to match the requirments of the scanner. By opening up the capture buffer, it is possible to save the text out to disk as a DV/80 and then use TI-Writer to edit, format, etc.

Last, TI XB did enable us to read the output of Omni-Reader. Here is the brief TEST program I wrote:

The following assumes that no line of printed text is more than 80 characters. The default on the 99/4A OPEN is Display Variable 80. However, for longer lines you can open up DISPLAY VARIABLE 255 files. A future article, part 4 in my running series on manipulating files, will cover how to convert such files to DV/80's.

First the program is written in plain English. After each line of English programming, is the equivilant XBasic programming.

(1) Clear your screen, print a message so the user knows the program has commenced, and open the RS232, configured to match the requirements of the scanning device.

```
100 CALL CLEAR :: PRINT "Start
Scanning": :
110 OPEN #1:"RS232.BA=1200.DA=8.PA=N"
```

2) Establish a keyboard scan with an "escape" so you can end the program when you are through scanning text or wish to quit.

```
120 CALL KEY(0,KEY,STATUS):: IF
STATUS<>0 THEN 150
```

3) Assume the user has activated the scanner - that is gone through the scanning procedure. This puts data into the RS232 buffer that the 99/4A establishes when you use the OPEN statement. Next, read the (input) data. If no data is found, go back and start again. If data is found, print it to the screen so we know we've been successful. Now go back and start over. (As text may contain commas, we cannot use the XB INPUT but must use LINPUT. INPUT stops reading when it comes to a comma.)

```
130 LINPUT #1:A$
140 IF A$="" THEN 120 ELSE DISPLAY
A$: : :: GOTO 120
```

(4) Last, CLOSE the open RS232 and END the program.

```
150 CLOSE #1 :: END
```

I'll wind up by saying, again, that even at a bargain price, OMNI-READER, was not a bargain. Al and I did have some 75 minutes of fun playing with it until we decided it was almost useless for any real volume of copying work. If we had paid $150 for O/R, that would have been $2.00 per minute! Quite expensive fun, yes?

As they said in the old Roman Arena: Thumbs Down! for OMNI-READER.
#################Eof################

## WISDOM LINES CONTINUED...

C. The Cell Pointer:
   1. Rectangular in shape
   2. Movement is controled by the direction keys. (Try It.)
      a. As the direction key is pressed, you will notice that the cell pointer moves about the screen depending on the direction selected.

D. The Status Line:
   1. Located at bottom of screen.
   2. Displays the location of the cell pointer and contents therein.
      a. This is also known as the active cell, where information can only be changed or modified in the highlighted cell location.

E. Scrolling The Worksheet:
   1. The screen shows only four columns at a time. Suppose that you need to move to column 23. To do so, you need to use the right direction key until that particular column is reached. Note that the pointer remains still, while the columns move right to left. The same procedure applies also to rows.

Well, this is the end of the first session of the Multiplan tutorial. I will continue with the "GOTO (G) COMMAND" next time. I will try to maintain this tutorial on either a bimonthly or quarterly basis, depending on my schedule. So, just get comfortable with Multiplan, and you'll be amazed at what this program can do for you.

- - - - - - - - - - - - - - -