

The Boston Computer Society

TI-99/4A User Group

Newsletter

SEPTEMBER 1989

Edited by Justin Dowling

LISTEN!

At the September meeting we will introduce a disk of the month feature which many other user groups do to reward meeting attendance and group participation. The first DOM is a disk full of utility programs you can use and modify at will.

For those of you who don't know, the BCS is chartered by Massachusetts as a non-profit institution like a university or a museum. Ours is not a simple user group organization like most of the groups who exchange newsletters with us. Consequently, exciting opportunities often come our way. Not only do you get more out of any opportunity than you put into it, but out of this latest opportunity we get a good meeting place, and the rent is in harmony with the BCS philosophy: don't pay for it; get it donated.

~~infocom, those crazy folks who made the hilarious and innovative~~ adventure games like *Hitchhiker's Guide to the Galaxy* or the *Zork* series donated their TI 99/4A to the BCS when they moved from Cambridge to the left coast. The system is in the backroom at One Center Plaza waiting for one of us to step forward, put on his/her systems engineer hat, and put it in a public school that will use it. (Ideally we would uncover a budding computer genius who would want to hack the 99/4a to pieces and in the process keep us alert and in good repair with his/her questions.)

Anyway, the front office located a budding museum that would love to have a live home computer exhibit that we could use on meeting nights. The Charles River Museum of Industry is located at 154 Moody St in Waltham. The Texas Instruments 99/4A, as you know, is one of the pioneer personal computers produced by the computer industry. The 99/4a is still alive! It is a computer for the rest of us when the industry is into Big Business. (A lot of technology has come along since then, but folks keep pulling rabbits out of their 99/4a hats by making their home computer do most of the things the big guy's equipment does in the office.

The Charles River Museum of Industry is right off 128. It is near a T train out of North Station. Nearby busses go to Watertown Square and to Newton. The Waban T station on the green line comes near it. The parking lot is over a footbridge that spans the Charles River where

it is still a narrow stream. The museum is wheelchair accessible should anyone bound to a chair wish to attend a meeting.

Think about what we can do for the museum and the senior citizen apartments next door; remember we usually get more than we give in such circumstances. Think about it and let us know.

The museum wants to make a live exhibit out of the 99/4a system. That probably means a patron would touch the keyboard to, say, draw a face with the facemaker cartridge plugged in. Better still! Save wear-and-tear on the cartridge port by dumping cartridges to disk and putting them on a Funnelweb menu to auto-load.

Telecommunications? Let one of the files on the Funnelweb menu be a communications program like TELCO with a macro to log the user onto one of our BBS's or another BCS BBS. Perhaps we can get a modem from the telecommunications user group when one becomes available. Do you think MYARC would want their Hard and Floppy Disk Controller (HFDC) displayed in a museum? Do you think they would want it bad enough to donate an HFDC card?

And what of the senior citizens who live nearby? You know, of course, that Sister Pat Taylor in an Iowa nursing home for nuns retired from schoolteaching for the most part, Sr. Pat Taylor has brought new meaning to a lot of lives, to people who find they are useful once again because they can make banners and cards and music and letters on one of Sr. Pat's 99/4a systems

Our group has been pencilled in to meet in the Museum on the third wednesdays of this November and December for starters. More on this later.... -EDITOR

C. COLUMN

by Donald L. Mahler

(Continued next page)

The Boston Computer Society
TI-99/4A User Group
One Center Plaza
Boston, MA 02108

This month I want to show some of the graphics functions possible with c99MDOS. These have changed quite a bit from those used with the original c99 GRF library. For example, we will use some enhancements for the SPHERE program I showed a few months ago. I find they work best if you load the MDOS patch from the C99MDOS disk before calling the program.

```

/* sphere for 9640 */
#include a:"stdio_h"
#include a:"float_h"
#include a:"video_h"
#define x 8 /* constants */
int i;
int pat1[]={0xFFFF,0xFFFF,0xFFFF,0xFFFF};
/* note the way pattern is input in blocks of 4 into an
array*/
/* this gives a a simple square we will use to form
letters*/
int n=4;int d=3;int z=7;
main()
{ grfmode(4,8); chrdef(136,pat1);
/* sets videomode to mode4- 256x212-16
colors;background color 8="cyan"*/
/* defined pattern is used as character "136" */
hcharc(7,2,136,32,9,5);
hcharc(17,2,136,32,9,5); vcharc(8,2,136,9,9,5);
vcharc(8,33,136,9,9,5); while (++z<=16)
hcharc(2,3,136,30,11,11);
hcharc(9,4,136,4,6,11);
hcharc(12,4,136,4,6,11);
hcharc(15,4,136,4,6,11);
vcharc(10,4,136,2,6,11);
vcharc(13,7,136,2,6,11);
vcharc(9,9,136,7,6,11);
vcharc(9,12,136,4,6,11);
hcharc(9,10,136,2,6,11);
hcharc(12,10,136,2,6,11);
vcharc(9,14,136,7,6,11);
vcharc(9,17,136,7,6,11);
hcharc(12,15,136,2,6,11);
vcharc(9,19,136,7,6,11);
hcharc(9,20,136,2,6,11);
hcharc(12,20,136,2,6,11);
hcharc(15,20,136,2,6,11);
vcharc(9,23,136,7,6,11);
hcharc(9,24,136,3,6,11);
hcharc(12,24,136,3,6,11);
vcharc(10,26,136,2,6,11);
hcharc(13,25,136,1,6,11);
vcharc(14,26,136,2,6,11);
hcharc(9,29,136,2,6,11);
hcharc(12,29,136,2,6,11);
hcharc(15,29,136,2,6,11);
vcharc(9,28,136,7,6,11);
/*hcharc and vcharc equivalent to hchar and vchar in
EXB */
locate(25,1);
puts(" Press any key ");
while(!poll(0));

```

```

bitclr();
txtmode(); /* back to text mode */
spher(); /*call function */
exit(0);]
/* spher() was printed in this column a few months
ago, so I will not
repeat it */

```

Or, if you prefer, we can dress up the SPHER with a sprite introduction: #include "a:video_h"

```

int i;
int pat1[]={0x00ff,0x8080,0xff01,0x01ff};
int pat2[]={0x00ff,0x8181,0x81ff,0x8080};
int pat3[]={0x0081,0x8181,0xff81,0x8181};
int pat4[]={0x00fe,0x8080,0xf880,0x80fe};
int pat5[]={0x00ff,0x8181,0xff84,0x8281};
/* five patterns as arrays; actual terms just like EXB
char def */
main()
{ grfmode(7,8);
/* 256x212-256 colors ; background CYAN */
sppdef(0,pat1); sppdef(1,pat2);sppdef(2,pat3);
sppdef(3,pat4);
sppdef(4,pat5);
/*define sprites */
sprite(1,0,14,1,1); sprite(2,1,11,10,10);
sprite(3,2,7,20,20);
sprite(4,3,13,40,40);sprite(5,4,16,50,50);
sprite(6,3,12,60,60);
/* spr#,character,color,dot position */
spmotn(1,20,20); spmotn(2,20,-20);
spmotn(3,8,-15); spmotn(4,-20,-15);
spmotn(5,0,-30); spmotn(6,-20,5);
spmag(2);
locate(25,1);
puts(" Press any key ");
while(!poll(0));
for(i=1;i<=6;i++)
spmotn(i,0,0);
sploct(1,40,15); sploct(2,40,40);
sploct(3,40,65); sploct(4,40,90);
sploct(5,40,115); sploct(6,40,140);
/* use sprites for title */
for(i=0;i<1000;+i)
{locate(25,1);}
puts(" Press any key ");
while(!poll(0));
spdall();
txtmode();
puts(" Now floating point for sphere!");
/* for actual program, substitute: "SPHER(); " */
exit(0); }
Again, to actually use program you would add SPHER
program at end!

```

INTRODUCTION TO THE UCSD P-SYSTEM BY RON WILLIAMS I/O CHECKING This month I will show you I/O checking that can be done to keep your program running if an I/O error occurs. Normally a UCSD

Pascal program will terminate if it detects an I/O error but by using a compiler directive to stop I/O checking it is possible for the program to take some action if there is an I/O error. A program I have put together demonstrates the use of program I/O checks the program will check to see if a file has been opened successfully if it has not a I/O error will be printed on the screen and the user will be prompted to try to input the file again. The program will not stop if an I/O error occurs, if the compiler directive I- was not used the program would stop and the p- system would re-initialize. The function ioresult is used if error checking is stopped and then the programmer is responsible for checking for errors normally a error message will be given but sometimes the program must do more to make the user of the program aware that a file read or write was not successful. The function ioresult is suppose to work with read and readln from the console but I have not got it to work this way. What I was trying to do was check for errors if I entered a character for a integer with readln I kept on getting error zero for this with error checking off error zero is no error output for ioresult. I even tried to use a small program from a book I have on the UCSD p-system but I could not get it work as stated in the book, this could be a error in the p-system for the 4A. But there are other ways of checking for errors from the keyboard I would enter everything in as strings and do error checking this way also there are many string functions in the p-system and some very good error checking can be done. I did an article on string data types a few months ago and if you look at that article you will get a lot of ideas for error checking this way. The codes that are returned by ioresult are in the form of integers and I like to convert them to strings using writeln so that they mean something to a person and not a computer. Each of the codes have a special meaning and that is where I use the case statement in the program below, in the UCSD Pascal manual is where the error codes returned are listed and what they mean. The program below is a sample program to check for file errors I think that you will find it quite easy to understand. PROGRAM errorcheck;

```
(* This program shows how to check for I/O errors *)
(* when opening a file for reading in data the *)
(* program will do an I/O check to see if the file*)
(* was opened successfully if it was not the *)
(* program lets the user try to input the file *)
(* again it then will check I/O again to see if *)
(* the file was opened successfully *)
(* The function errorstatus has other uses also *)
(* it could be used as a file I/O check for *)
(* rewrite and for other types of file I/O errors *)
```

```
var
  answer : char;
  fname : string;
  pfile : text;
  eflag : boolean;
```

```
function errorstatus : boolean;
var
  errornum : integer;

begin
  errornum:=ioresult;
  if errornum > 0
  then
    begin
      gotoxy(1,20);
      errorstatus:=true;
      eflag:=true;
      case errornum of
        1 : write('bad block');
        2 : write('illegal device number');
        3 : write('illegal i/o request');
        4 : write('i/o operation cancelled by user');
        5 : write('volume went off-line');
        6 : write('file lost in directory');
        7 : write('bad file name');
        8 : write('no room on volume');
        9 : write('volume not found');
        10 : write('file not found');
        11 : write('duplicate directory entry');
        12 : write('file already open');
        13 : write('file not open');
        14 : write('bad input information');
        15 : write('ring buffer overflow');
        16 : write('write protect');
        17 : write('illegal block');
        18 : write('illegal buffer');
      end;
      if (errornum > 18) or (errornum < 1)
      then
        write('error *=',errornum);
      end
    else
      errorstatus:=false;
      gotoxy(1,22);
      write('Press any key');
      read(answer);
    end; (* errorstatus *)
```

```
begin
  repeat
    eflag:=false;
    page(output);
    gotoxy(1,1);
    write('enter file name=>');
    readln(fname);
    (*$1-*)
    reset(pfile,fname);
  until not errorstatus;
  (*$1-*)
  if not eflag
  then
    begin
      gotoxy(1,20);
      write('File open ok');
```

```
close(pfile,lock);
end;
gotoxy(1,22);
write('Press any key');
read(answer);
page(output);
end.
```

(The following article is reprinted from PUNN)

FRACTAL EXPLORATION
by Mark Schafer

The editor asked me to write an article on fractals, the program FRACTAL EXPLORER, and Version 3.0 This is to enlighten people still in the dark about fractals and to let people know about the latest version of a program that generates them.

First, what is a fractal? Those of you who already know can skip this part. Fractals (from the Latin "fractus" meaning broken up) are shapes that are so infinitely detailed that they are said to occupy a "fractional dimension". On the surface it appears that they occupy only two dimensions (or whatever). But for an ordinary two-dimensional object, if you magnify it, you see more of the same. Magnify a fractal and you see detail you didn't see before no matter how close you get. ~~It is this property that gives it part of another dimension, but not a whole dimension since its depth is immeasurable.~~

It is said that some structures in nature emulate fractals such as coastlines, trees, and clouds. The science that studies fractals, fractal geometry, looks at ways that such structures can be described or simulated by using fractals. Such simulations can look remarkably lifelike.

But all that is beyond the scope of this article. I am going to guide you through a specific fractal known as the Mandelbrot Set, named after Benoit B. Mandelbrot, the man that founded fractal geometry.

The Mandelbrot Set is a set of complex numbers, so to understand it you must first get a grasp of complex numbers. They are so called because they have two parts--a real part and an imaginary part. The real part is from the set of real numbers, all the numbers found on a number line. The imaginary part contains the square root of negative one. Negative numbers don't have square roots (since no real number squares to be negative) so it is called an imaginary number. This value is assigned the name, i.

Now i has a few characteristics that must be noted. What happens if you square i? Since i is the square root of negative one, its square must be negative one. What happens if you cube it? Then you get negative i. Lastly, if you take i to the fourth power, you get 1. Using i,

you can write any other imaginary number. For instance, how would you write the square root of negative 64? That could be written as 64 times -1, the square root of 64 is 8, the square root of -1 is i, so the answer is 8i.

So a complex number would look something like 3+8i, where 3 is the real part and 8i is the imaginary part. Other examples would be 7-2i, -3.4+9i/6i, and -2/7-.0025i. Now let's do some complex math! To add two complex numbers, simply add their corresponding parts (real to real, imaginary to imaginary). This is easy, right? Subtraction is similarly done. So for examples:

$$\begin{array}{r} 3+8i \\ + 2-1i \\ \hline 5+7i \end{array} \quad \begin{array}{r} 7-.5i \\ -6+3.5i \\ \hline 1-4i \end{array} \quad \begin{array}{r} -4.1+2i \\ + -2.6-5.5i \\ \hline -6.7-3.5i \end{array} \quad \begin{array}{r} -7.3-6.25i \\ -3.6-2.12i \\ \hline -10.9-4.13i \end{array}$$

Multiplication is much more complicated, though. Think of it as multiplying one two digit number by another where the first digit is real and the second is imaginary. You know, you have to multiply each digit in one number by each digit in the other and add the results. However, remember to bear in mind that 1 times i is i. Here's a comparison of regular multiplication and complex multiplication and look at the similarity:

$$\begin{array}{r} 38 \\ \times 57 \\ \hline 266 \\ 190 \\ \hline 2166 \end{array} \quad \begin{array}{r} 3+8i \\ \times 5+7i \\ \hline 21i-56 \\ 15+40i \\ \hline 2166 \quad 15+61i-56 = -41+61i \end{array}$$

There's an extra step on the complex one to simplify the answer to two parts, but notice there is no carrying.

Division is much more complicated, but luckily generation of the Mandelbrot Set does not demand complex division, so I'm not going to go into it here.

The home of the real numbers is the number line; so where is the home of complex numbers? The complex plane. Imagine that coordinate system you should've learned in school where the center is (0,0), 1 unit to the right and 1 unit up is (1,1), and 1 unit to the right and 1 unit down is (1,-1) and so on. Things can easily be converted to a complex plane by thinking of x (the first number) as the real part and y (the second number) as the imaginary part. So instead of the x-axis and y-axis, we have the real axis and imaginary axis. Notice the real axis is the number line. So 3+8i is located 3 units to the right and 8 units up; similarly -6.7-7.1i is located 6.7 units to the left and 7.1 units down.

This series will be continued next month when space permits. Stay tuned! -ED

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 R0,>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 >2010
VMBR EQU >202C
MYWS BSS 32
H30 BYTE >30
BYTE 11 String Length
DSK BSS 11 String Save Buffer
EVEN
TRACK LWPI MYWS
LI R0,>3FF5 Address of Drive #
LI R1,DSK
LI R2,11
BLWP @VMBR Drive # and Filename
AB @H30,@DSK Drive # to ASCII
CLR R0 Not an Array
LI R1,1 First Parameter
LI R2,DSK-1 Length Byte
BLWP @STRASG Assign to A#
LWPI >83E0 GPLWS
CLR @>837C Clear STATUS
B @>70 Return to XBasic
END
```

```
10 CALL CLEAR :: CALL SCREEN(
5):: FOR I=0 TO 12 :: CALL CO
LOR(I,16,1):: NEXT I
20 CALL INIT :: DELETE "LD-CH
DS" :: A0=RPT*( " ",10)
30 CALL LINK("VPEEK")(16373,0
,N,A#)
40 A#="DSK"&STR$(N)&". "&A#
50 DISPLAY AT(0,5):"My name i
s now": :TAB(5);A# : :TAB(
5);"Rename me and run me": :
:TAB(5);"in another drive."
60 CALL KEY(0,K,S):: IF S=0 T
HEN 60
```

UGOC
BULLETIN
BOARD
300/1200/2400 BAUD
(714) 751-4332

Asgard Rebuttal

By Chris Bobbitt

Peter Hoddie's review of *Asgard News* in the "Meeting Newsletter August and a half 1989" contained a number of factual errors which are no doubt based on only a passing acquaintance with the publication in question (which he admits in his article) and of myself.

(1) My last name is indeed spelled "Bobbitt" - with 2 "T"s. I don't know why - I've never asked my father.

(2) *Asgard News* is not a *MICROpendium* supplement. True, we try to publish material not found in *MICROpendium*. If we didn't, no one would subscribe to our magazine! This only makes us as supplemental as any other publication that publishes information about the 99/4A not found in *MICROpendium*.

(3) *Asgard News* is far from "promotion, propaganda, and expensive advertising". Approximately 25% of the 1st issue was devoted to Asgard related items (including the sections on Bug Reports and New Versions - which we also publish in *MICROpendium*, all ads and announcements and tutorials), 18% of the 2nd issue, 25% of the third (mainly due to a lengthy *Press*TM description), 13% of the 4th and 18% of the fifth issue. That averages out to about 20% of the magazine. Interestingly enough, at least 1 of the 5 pages of the newsletter I obtained was devoted to Genial products and Peter's plans for products (and that isn't counting the page he used in his review of *Asgard News*, or the 1/2 page he used in discussing his change of residence). That is also 20% of the newsletter.

We have an 85% renewal rate - if our subscribers didn't like what they were reading, they would certainly not re-subscribe.

(4) We can't afford to give away this information - otherwise we would. Many of the big software companies in the PC and Mac worlds (some of the ones mentioned by Peter have over \$1 Billion in sales) do give away magazines, but then again we aren't a million, much less a billion dollar concern, and we don't charge hundreds of dollars for our products (and hence could afford to give away a magazine subscription as part of the deal).

We try to make the magazine worthwhile by publishing quite a bit of non-Asgard information (usually about 80% of the magazine). In the last five issues, we've published dealer, user group, and hardware and software vendor lists (with Genial Computerware's address and name), and highly regarded articles like "The New 99/4A". We've also published tutorials on databases and telecommunications in general, on TI-Artist and TI-Base in particular (both *Texaments* products). We've described expansion options for our cassette readership, continued an extensive Geneve Corner that Beery Miller of *9640 News* found worthwhile, publicized user groups, and published information about the current state of user groups in general. We've also published reader mail and provided answers to questions on many topics. Most of our mail has ranked *Asgard News* with *MICROpendium* as one of the best TI-99/4A publications (which we think is excellent company) - and we are proud of that.

(5) Over the last five issues we've published announcements for: *Fortran 9640*, *Databiotics* products, *Telco*, *Genial Traveler*, *Funlplus*, *Turbo Pasc'99*, *Archiver 3.0*, *Rave* products, *Dijit* products, *Bud Mills*, *Triton*, *Texaments* products, *CIS*, *Art Green*, *Delphi*, *Mechatronics* and others. We've also had announcements for *MacFlix* and the *MDOS Development Package* (I heard no objections to either of those from Peter), and gave a free ad to *Genial Traveler*. We've never had an announcement for *Graphics Expander* because we only publish announcements for new products, and that product was released in 1987 - about 6 months before the first issue of *Asgard News* was delivered. I don't know what *AV-Indexer* is and what it is designed to

compete against - send me new product announcements (as Texaments, which certainly competes with us, does) and maybe we'll publish them. Announce a word processor (though Press is a trademark), and we'll happily publish the announcement.

(6) Our article on the M-DOS Development Package was written at the end of June. At that time, we had seen no announcement on any of the Telecommunications networks we frequent (GENie and CIS) regarding the status of the program - which had been slated for release in April. It is a vendors duty to make these things known to customers, not the customers duty to get in touch with the vendor to find out. Within 2-3 months of the delay for Press™ we had sent postcards out to customers informing them of the delay and giving them an option of a full refund, and had articles on most computer networks and in *MICROpendium*.

(7) The implication that our articles were somehow written because we were not given the opportunity to market the *MDOS Development Kit* (or *GenProg* as it is now called), is incorrect. In a recent conference with Myarc on Compuserve, I was trying to make the point (unsuccessfully and not altogether succinctly) that Myarc should be the only one writing a developer's package for the Geneve. I don't believe *any* third-party firm (including Asgard) should do so. The reason for this is simple: it is not in the best-interest of any third-party firm to release this kind of information with any haste. Any third-party firm entrusted with writing and marketing such a package would be strongly tempted to hold onto the information for as long as possible - because it gives them an insurmountable advantage over every other firm developing 9640-only software.

(8) Documentation for Asgard products has consistently rated well with most reviewers - with the exception of Barry Traver, who was a partner in Genial Computerware when he reviewed our products in *Computer Shopper*). *Legends* received an "A" for documentation in the Nov.'87 issue, *Artist Enlarger* an "A+" in July '88, *Page Pro 99* a "B+" in July '89, *Artist Borders* four stars in April '89, *EZ-Keys* an "A" in Jan.88, *Calendar Maker 99* a "B" in April '88 (in the same issue *PC-Transfer* also got a "B" for documentation), and so on. I wrote all of those manuals. ~~Asgard Software products have consistently rated highly in documentation, and in packaging as well as documentation they have become the standard by which other vendors products are rated in many cases.~~

(9) Press™ is late, definitely - but we've never made a secret of it. Our two-page ad was placed when we believed the product would be available on schedule - and we never placed another after it became clear the program would be late. We've since learned our lesson - we don't talk about products any more until they are ready to be shipped. The only other time we were late with a product was *Font Writer II*, which had been promised by the author in March and was delivered in September of 1987. Finally, how can Peter assault me alone for being late with Press since Charles Earl (the author of the program) also accepts responsibility for the delay?

(10) I apologize for saying you graduated from Boston College - my mistake. I knew it was *some* school in Boston.

(11) In our original article, when we stated "no word is available at this time of future plans for Genial", again that was at the end of June and we had heard nothing, and even Peter's partner didn't know at the time. Peter neglected to mention that we gave Genial Computerware's full address afterwards and stated it was where customers could write to order and for product support. Any errors in detail were inadvertent and will be corrected in the next issue of *Asgard News*.

(12) I meant no slur against the residents of Massachusetts (particularly since I once was one) - I was referring in fact to the problems the Rt.128 computer industry has had recently, which Apollo being purchased by HP, Wang in serious trouble, and some parts of DEC relocating. It

makes sense for any new engineer to go where the jobs are, and the jobs are going to Silicon Valley at the moment. Peter is to be congratulated for his willingness to relocate.

(13) Finally, we publish the "Rumors" column because it is the most popular part of the magazine. We've had several scoops over other publications since we started it - including the story about Tenex leaving the TI world, and Triton being sold to Activision/Mediagenics among others. We've only been "wrong" twice out of over 30 (and in both cases the disagreement was over minor details and not the focus of the article) - I think that says good things about the grapevine we listen to.

Petty remarks about my person certainly don't befit the owner of a software company that wants to become a major player in the TI software industry.

Our articles, which I've reproduced in their entirety below, were printed in good faith. The only part of any of them I regret was the statement "how long can a manual take to write" - that was editorializing and not reporting and I apologize. The rest I feel is accurate and fair within reason, and I invite our readers to be the judge:

M-DOS Development Package Late

The M-DOS Development Package by Paul Charlton, and announced for release in April by Genial Computerware, still hasn't materialized as the beginning of July approaches. No word is available when it will be making its appearance, but rumor has it that only the documentation is awaiting completion. But then, we first heard this rumor a month ago - how long can a manual take to write?

Genial Relocates?

The major partner of Genial Computerware, J.Peter Hoddie, has announced that he is re-locating to the San Francisco area. A recent graduate of Boston College, Mr. Hoddie is evidently taking his newly minted Engineering Degree to employers in the thriving Silicon Valley area, away from ailing Massachusetts (which is suffering record budget deficits). No word is available at this time of his future plans for Genial, and the status of his partnership with co-owners Corson Wyman and Barry Traver. The current address for ordering and software support of Genial Computerware is P.O. Box 183, Grafton, MA 01519. We will of course keep you updated of any new addresses as they become available.

Thank you.