CEDAR VALLEY 99er UG   AUGUST 1993

PRESIDENT:	Jack Johns	(319) 366-4541
VICE PRES:	Bob Wahlstrom	(319) 393-6042
TREASURER:	Bruce Winter	(319) 393-0610
SECRETARY:	Wayne Betts	(319) 377-2493
NL EDITOR:	Gary Bishop	(319) 377-9574
LIBRARIAN:	Bob Heiderstadt	(319) 927-4215

CV99 UG99 IOWA

CEDAR RAPIDS/MARION

Supporting the TI-99/4A and 9640 in Eastern Iowa for over 10 years!

NEXT MEETING: 6:30 PM August 10, 1993

WEST MUSIC, COLLINS ROAD PLAZA

CONTENTS

Our July meeting started out by swapping horror stories about the great flood of '93. Most members in attendance stayed dry, but we were not in a stable situation. Several of the usuals weren't there, and everyone hoped it wasn't due to flooding. Many detours were necessary due to closed roads and interstates. After obtaining a console, disk copying and investigations ensued.

From Gary Bishop: This will be my last newsletter. I need someone else to take over the reins. The next issue already has part 3 of the 9938 tutorial printed out, so September's issue shouldn't be too hard. I will keep the next editor well stocked with articles and ideas. Give it a shot.

SUMMERFEST '93

Sunday August 15, 1993
5000 "J" Street S.W. Cedar Rapids, Iowa 52404

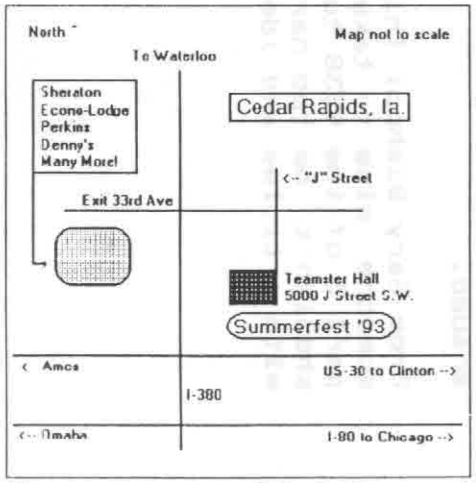
Iowa's Premier Hamfest sponsored by the Cedar Valley Amateur Radio Club and sanctioned by the Amateur Radio Relay League.
Over $2500 in prizes and gifts!

Air Conditioned Exhibit Hall
Free Coffee!
Consignment Table
Huge outdoor flea market
Tickets - $4.00 at the door
(no advanced ticket sales - sorry)

VE Exams (walk-in OK)
(For pre-registration contact)
WVØC Bob Brus
811 Williams Drive
Marion, Iowa 52302
319.373.9628

Table rental - $11.00 each
Table rental contact Wayne Hughs
Attn: Table Rental
3109 6th Street
Marion, Iowa 52302
319.373.2391

For more general information on the hamfest contact KEØMS Wayne Harrah @ 319.848.7481 or KEØMS@WAØRJT.IA

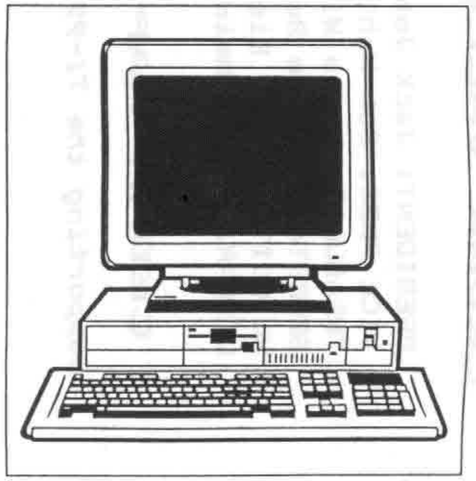


Easy access and lodging and fine dining!!!

Hamfest		*Exams **	
6:00	*Grounds open for Flea Market setup*	*10:00*	*Registration*
7:00	*Hall open for Vendor setup*	*10:10*	*20 WPM Code*
8:00	*Hall OPEN!*	*10:30*	*13 WPM Code, Novice Written*
10:00	*EIDXA Forum*	*11:00*	*5 WPM Code, Tech Written*
11:00	*Repeater Council*	*11:30*	*General Written*
12:00	*ARRL Forum*	*12:00*	*Advanced Written*
1:00	*AMSAT Forum*	*12:30*	*Extra Written*
2:00	*Prize Drawing*		
4:00	*Hall Closes*		

** Times are Estimated*

Summerfest '93
Schedule of Events



Ham and Computer Vendors
Something for everyone!!!

THE FACTS ABOUT TI MEMORY SYSTEMS
AN EDITORIAL

Over the last six months there has been a lot of noise on the computer networks and in user group newsletters on issues related to extended memory cards for the 99/4A. Some people have blatantly asked people to come out and choose sides on a very complex issue without understanding what they are choosing. In fact, what should be a pretty objective decision has been turned into an emotional gut-churner - a question decided by loyalties, petty rivalries, lies and innuendo. Frankly, this is why we are in the situation we are in today - and why companies like Myarc and Corcomp left the community. Everything technical debate gets turned into a personal vendetta, thinly disguised ambition is allowed to prevail over substance, and the community eats its young yet again.

When I set out almost 3 years ago with a really talented bunch of guys to put together a new kind of memory card for the 99/4A, I had believed that the community had finally outgrown that kind of thing. I was wrong. Seeing all of this stuff all over again has made me seriously consider throwing in the towel once and for all.

Why? Because everything we've done with these cards has either been ignored, mis-represented, or labeled as too "controversial" or "not ready for prime time".

For 8 months we've been mailing out press releases, articles and newsletters about our memory cards that apparently no one is reading, and user groups aren't re-printing or even reporting on. The only reference to what we've done that I ever see in print is usually in an article about our competitors, or in an editorial that simply says that there has been a "debate" and that it has "gotten out of hand".

This is simply ridiculous.

This is the most important thing I've been involved with in the 10 years I've been in this community, and unless the community gives this a fair hearing, well, I guess it's finally time to cut my losses.

Here is my last attempt to get the unvarnished facts out in front of you, the reader.

1. The Asgard Memory System (AMS) is available NOW - it is NOT still in "development". We announced the product the day it was commercially available for sale, and in stock. In the last 8 months, we've been refining the product, writing software, and working on the next generation card. Our only competitor announced their product over a year ago, and have yet to release more than press notices (which all seem to be faithfully reprinted everywhere). It is pretty hard to compete against something that so far exists only on paper - especially when the unreleased product gets more press than the one that you can buy today!

2. We started AMS almost 3 years ago - long before there ever was a "National Committee for TI Standards". This so-called committee has never met more than once, doesn't include most of the TI hardware or software developers in the U.S., much less the rest of the world, and has produced a specification for memory systems without any real debate, which endorses our competitor's plans. Before we had a chance to object, it was the declared "standard". Can you say "railroaded"?

3. Our memory system was designed to the only standard TI ever made for extended memory on the 99/4A - the one used in the TI-99/8. In fact, the guy who DESIGNED the TI-99/8 said our design was identical to the one TI specified.

4. Because our design was built to TI's specifications, it doesn't conflict with any other card in the P-BOX - except a 32K card. You can plug it in and your Horizon RAM-disk, Myarc HFDC, or anything else you have will still work fine.

5. Our design uses standard, off-the-shelf components. EVERY other extended memory design uses lots of custom ICs, and even more custom programming (as in a big DSR). Custom parts not only drive up the development time, they also drive up the cost, and guarantee that the design remains proprietary. By using off-the-shelf parts, we keep the price down, and guarantee competition. Remember how much TI used to charge for the 32K card when they were the only one making them?

We designed our system to the "KISS" method - "Keep It Simple, Sam".

THE FACTS ABOUT TI MEMORY SYSTEMS
AN EDITORIAL (Continued)

6. Our system is tried and tested. We use the exact same "memory mapper" (the chip that controls the computers use of memory) that TI used in their 99/8, their 9900 minicomputers, and that IBM used in the very first IBM PCs. This component has been available for 10 years - all "bugs" in it have long been removed.

7. Everything about our system is "open". Anyone can write a program for it or enhance it - the hardware and software specifications are available free of charge. Heck, the 5-disk development system we've spent the last 18 months writing is even fairware - and posted on the bulletin boards.

8. The AMS is very fast. It can switch pages over 10 times faster than any competitor, and with little program code (even in Assembly). Why is speed important? If you are sorting 512K of data, or loading 512K of pictures, you'll notice the speed - in fact, you'll notice the other system is less than half the speed.

9. Our system doesn't have its software in a DSR - and we are proud of it! Why?

   A. We found that putting the operating software in a DSR makes it run much slower than if it was in RAM - and really doesn't give any benefit to the programmer or the user.

   B. Any DSR increases the chance for compatibility problems - who wants to waste time finding problems with Myarc cards?

   C. A DSR is "fixed". If you find a bug in it, the only way to correct it is to replace it. Consider all the pain Myarc users have gone through with EPROM upgrades of the HFDC and the Geneve.

   D. If programs are written to work around a DSR bug, they may not work when the DSR is fixed.

   E. If the software to use the card is built into each program, than the only thing we have to do to correct a bug is issue an upgrade. Old programs written for earlier versions of our operating system software would continue to work fine, and new programs could take advantage of new features without worrying about hardware compatibility problems - since the operating system isn't in hardware.

   F. Why do you think Microsoft and Apple load their operating systems from disk, and not from ROM chips?

10. We have a complete set of development tools available NOW. Even if our competitors released their card today, it would be a year before they had a system that was as easy as ours is for programmers. Because our software was designed before our hardware, we were able to design a "programmer friendly" system that is far easier to program than any other extended memory system. This is important - as so many people have said, who wants a memory card there are no programs for?

In the last 8 months since we released our first AMS card we've released 2 software packages that take advantage of the card (including the word processor FIRST DRAFT), and software from other people has started to appear. Around 20 AMS cards are in the hands of developers around the world.

Is any of this news? Apparently not - I've seen few of the facts above in print anywhere, even though we've put them in a half-dozen articles.

The facts, on their own merit, should be compelling enough for people to put aside their differences and really weigh the benefits of what we've done - instead of consigning it as some "curiosity", or ignoring it.

We wanted to put together something that was cheap enough to build that every TI user could have one, and yet was simple enough to write programs for that every TI programmer could do so. I think we've done that. If the TI world isn't interested at this point, doesn't care, or wants to keep waiting for fantasies, well, I can take a hint.

Thank you.

Chris Bobbitt
July 2, 1993

```
Article 1004 of comp.sys.ti:
Newsgroups: comp.sys.ti
Path: zodiac.cca.cr.rockwell.com!moe.ksu.ksu.edu!zaphod.mps.ohio-state.edu!howla
nd.reston.ans.net!bogus.sura.net!news-feed-1.peachnet.edu!umn.edu!vx.cis.umn.edu
!daven
From: daven@vx.cis.umn.edu (David Nieters)
Subject: V9938 Graphics Mode 4 Tutorial Part 2 (repost)
Message-ID: <19APR199310514986@vx.cis.umn.edu>
News-Software: VAX/VMS VNEWS 1.41
Sender: daven
Nntp-Posting-Host: vx.cis.umn.edu
Organization: University of Minnesota CIS
Date: Mon, 19 Apr 1993 15:51:00 GMT
Lines: 346
```

In part 1, we saw a program that would draw lines on the screen in multiple colors. To do this, we had to plot each point by calculating a memory address, reading the contents of that location, and storing the color of the dot we wanted to display.

In this part, we are going to tell the 9938 where to put the dot and what color to make it. The 9938 will then do the work of determining the correct memory address to store the color value and making sure it is properly put in either the high or low nybble. Before we see the new source code, we will review some of the command registers of the 9938 that make this possible.

Registers #32 through #46 are used by the 9938 for executing commands. The ones we will use are #36-#39, #44 and #46. Their usage is as follows -

R#36 - Lower 8 bits of the X coordinate
R#37 - Higher 1 bit of the X coordinate
R#38 - Lower 8 bits of the Y coordinate
R#39 - Higher 2 bits of the Y coordinate
R#44 - Color of the point
R#46 - Command register

Since our screen is 212X256, the high bits of the X and Y coordinates will always be zero. Therefore, R#37 and R#39 are always zero. Also, since we have only 4 bits per color in Graphics 4 mode, the lower 4 bits of R#44 contain the color and the higher 4 bits contain zeros.

The upper 4 bits of the command register (R#46) tell the 9938 what operation we want to perform. To plot a single point, this value is 0101 binary. The lower four bits tells the 9938 what we want to do with the point that is already on the screen. In our case we want to replace the existing point with the new point, so we put in a value of 0000 binary.

Once you write to R#46, the command gets executed by the 9938. Therefore, it is necessary to write data to all the other registers before writing R#46. The 9938 will take a finite amount of time to perform a command before it can be ready to execute another. The 9938 provides a status bit in one of the status registers to let you know if it is ready to accept a new command yet. I've found in this program that the 9938 can plot a point faster than the 9900 can compute where the next point will be, so I do not check the status bit. We will see in Part 3 where a command will take a sufficient amount of time that we may have to wait for it to complete before trying to execute another.

We have now turned our 99/4A into a parallel processor by getting the 9938 to perform one task while the 9900 performs another. The speedup isn't that noticable (in my opinion) over the program in part 1. In part 3, however, we will get the 9938 working more and realize some serious performance gains over what we have done so far.

The source for part 2 follows. It is identical to that from part one, except for the routine POINT now sets the command registers to plot the point rather than directly writing to VDP memory. I have also changed the register usage slightly since R0 must be used in the VWTR routine.

```
        REF   VWTR,VSBW,VMBW,KSCAN,VSBR
        REF   VDPWD,VDPWA,VDPSTA

HEIGHT EQU   212          NUMBER OF LINES
NUMLIN EQU   100          NUMBER OF LINES WE DRAW BEFORE ERASING SCREEN

* CLEAR THE SCREEN
*
* THIS ROUTINE CLEARS THE SCREEN BY WRITING ZEROS IN THE
* PATTERN NAME TABLE.  WHEN DEALING WITH THE LARGER MEMORY
* SPACE OF THE V9938, WE HAVE TO BE SURE THAT REGISTER #14
* IS CLEARED BEFORE WE START.  OTHERWISE WE MIGHT BY ZEROING
* OUT HIGHER AREAS OF MEMORY THAN WE WANT TO.
*
CLEAR  LI    R0,>0E00      RESET OUR VDP ADDRESS
       BLWP  @VWTR
       LI    R0,>0040
       MOVB  R0,@VDPWA
       SWPB  R0
       MOVB  R0,@VDPWA
       LI    R2,HEIGHT8  WE WILL WRITE 24,576 ZEROS
       CLR   R0
CLEAR1 MOVB  R0,@VDPWD
       DEC   R2
       JNE   CLEAR1
       RT

* RANDOM NUMBER GENERATOR
*
* THIS PROCEDURE RETURNS A (NOT SO) RANDOM NUMBER IN R1.
* IT ENSURES THE RANDOM NUMBER WILL NOT BE 0.
*
RAND   MOV   @SEED,R1
RAND1  AI    R1,>1D6B
       JEQ   RAND1
       MOV   R1,@SEED
       RT

SEED   DATA >690A

DX1    DATA 0           THESE LOCATIONS ARE USED TO STORE
DX2    DATA 0           HOW FAR THE ENDPOINTS MOVE EACH
DY1    DATA 0           TIME A LINE IS DRAWN
DY2    DATA 0

* COLOR FLAG
*
```

```
* WHEN COLOR FLAG IS ZERO, THE LINES WILL APPEAR IN
* DIFFERENT COLORS.  WHEN IT IS NOT SET TO ZERO, ALL
* LINES WILL BR DRAWN IN THE SAME COLOR.  IT'S TOGGLED
* BY PRESSING THE 'C' WHILE LINES ARE BEING DRAWN.
*
CFLAG  DATA 0

* POINT
*
* POINT WILL TAKE AN X COORDINATE IN R9 AND A Y
* COORDINATE IN R10 AND A COLOR IN R2 AND PLOT THAT
* POINT ON THE SCREEN
*
POINT  LI   R0,376     CLEAR VDP REGISTER #37
       BLWP @VWTR
       LI   R0,396     CLEAR VDP REGISTER #39
       BLWP @VWTR
       MOV  R9,R0
       AI   R0,366     SET THE X COORDINATE IN R#36
       BLWP @VWTR
       MOV  R10,R0
       AI   R0,386     SET THE Y COORDINATE IN R#38
       BLWP @VWTR
       MOV  R2,R0          SET COLOR REGISTER
       AI   R0,446
       BLWP @VWTR
       LI   R0,466+>50 SET THE COMMAND REGISTER
       BLWP @VWTR
       RT

* PLOT
*
* THIS ROUTINE PLOTS A LINE FROM (X1,Y1) TO (X2,Y2)
* THESE COORDINATES ARE LOCATED IN THE CALLERS
* REGISTERS R6,R7,R8 AND R9.  THE COLOR IS
* SPECIFIED IN THE CALLER'S R10.
*
PLOT   DATA >8300
       DATA PLOT1

PLOT1  CLR  R12
       LI   R5,1
       LI   R6,1
       MOV  @16(R13),R7
       MOV  @12(R13),R9
       S    R9,R7
       JLT  PLOT11
       JMP  PLOT2
PLOT11 NEG  R7
       NEG  R5
PLOT2  MOV  R7,R7
       JNE  PLOT3
       SETO R12
PLOT3  MOV  @18(R13),R8
       MOV  @14(R13),R10
       S    R10,R8
       JLT  PLOT4
       JMP  PLOT5
```

```
PLOT4   NEG   R6
        NEG   R8
PLOT5   MOV   @10(R13),R2  GET COLOR
        BL    @POINT
        C     R9,@16(R13)
        JNE   PLOT6
        C     R10,@18(R13)
        JNE   PLOT6
        RTWP

PLOT6   MOV   R12,R12
        JLT   PLOT7
        A     R5,R9
        S     R8,R12
        JMP   PLOT5
PLOT7   A     R6,R10
        A     R7,R12
        JMP   PLOT5

* MAIN PROGRAM
*
START   LWPI >8320
        LI    R2,VDPREG     SET VDP REGISTERS
L1      MOV   *R2+,R0
        JLT   L2
        BLWP @VWTR
        JMP   L1

L2      BL    @CLEAR        CLEAR THE SCREEN

        CLR   @CFLAG
        CLR   R3            R3 COUNTS THE NUMBER OF LINES WE HAVE DRAWN

        LI    R6,>80        SET THE ENDPOINTS FOR OUR FIRST LINE
        LI    R7,>60
        LI    R8,>D3
        LI    R9,>13

        CLR   R0            SET THE INITIAL AMOUNTS THE ENDPOINTS
        INCT R0             MOVE BY
        MOV   R0,@DX1
        INCT R0
        MOV   R0,@DY1
        INCT R0
        MOV   R0,@DX2
        INCT R0
        MOV   R0,@DY2

LOOP    MOV   @CFLAG,R0
        JNE   L5
        BL    @RAND        PICK A RANDOM COLOR
        ANDI R1,>F
        MOV   R1,R5
        CI    R5,2         MAKE SURE WE DON'T HAVE BLACK
        JHE   L5
        ORI   R5,2
L5      A     @DX1,R6      MOVE THE ENDPOINTS
        A     @DY1,R7
        A       @DX2,R8
        A       @DY2,R9

* CHECK TO MAKE SURE THAT NO ENDPOINTS HAVE MOVED OFF
* THE SCREEN.  IF SO, REVERSE ITS DIRECTION.
*
        MOV     R6,R6
        JLT     L6
        CI      R6,>100
        JLT     L7
L6      NEG     @DX1
        A       @DX1,R6

L7      MOV     R8,R8
        JLT     L8
        CI      R8,>100
        JLT     L9
L8      NEG     @DX2
        A       @DX2,R8

L9      MOV     R7,R7
        JLT     L10
        CI      R7,HEIGHT
        JLT     L11
L10     NEG     @DY1
        A       @DY1,R7

L11     MOV     R9,R9
        JLT     L12
        CI      R9,HEIGHT
        JLT     L13
L12     NEG     @DY2
        A       @DY2,R9

L13     BLWP @PLOT

L14     CLR     R0              CHECK TO SEE IF A KEY IS PRESSED
        MOVB R0,@>8374
        BLWP @KSCAN
        MOVB @>8375,R0
        MOVB @>837C,R1
        JEQ     L16
        CI      R0,>0500        CHECK FOR QUIT KEY
        JNE     L15
        B       @QUIT
L15     CI      R0,>4300        CHECK FOR "C" KEY PRESSED
        JNE     L14
        INV     @CFLAG          TOGGLE THE COLOR FLAG
L16     CI      R0,>FF00
        JNE     L14
        INC     R3
        CI      R3,NUMLIN       SEE IF WE HAVE MORE LINES TO DRAW
        JNE     LOOP            IF SO, GO BACK AND DRAW THEM

        CLR     R3
        LI      R2,10
        LI      R4,>FFFF
DLY     DEC     R4              WAIT A LITTLE BEFORE CLEARING THE SCREEN
```

```
        JNE   DLY
        DEC   R2
        JNE   DLY

        BL    @RAND         COMPUTE NEW RANDOM MOVEMENTS
        MOV   R1,R1
        JLT   L17
        ANDI  R1,7
        JMP   L18
L17     ORI   R1,>FFF8
L18     MOV   R1,@DX2
        BL    @RAND
        MOV   R1,R1
        JLT   L19
        ANDI  R1,7
        JMP   L20
L19     ORI   R1,>FFF8
L20     MOV   R1,@DY1
        BL    @RAND
        MOV   R1,R1
        JLT   L21
        ANDI  R1,7
        JMP   L22
L21     ORI   R1,>FFF8
L22     MOV   R1,@DX1
        BL    @RAND
        MOV   R1,R1
        JLT   L23
        ANDI  R1,7
        JMP   L24
L23     ORI   R1,>FFF8
L24     MOV   R1,@DY2

        BL    @CLEAR        CLEAR SCREEN
        B     @LOOP         START OVER

QUIT    LI    R2,REG2       RESTORE VDP REGISTERS BACK TO NORMAL
QUIT1   MOV   *R2+,R0
        JLT   QUIT2
        BLWP  @VWTR
        JMP   QUIT1
QUIT2
        LIMI  2
        BLWP  @0

* VDP REGISTERS TO SET VDP TO GRAPHICS 4 MODE
*
VDPREG  DATA >0006
        DATA >0160
        DATA >021F          LOCATE NAME TABLE AT ADDRESS 0
        DATA >0711          SET BACKGROUND TO BLACK
        DATA >080A          INHIBIT SPRITES
        DATA >0980          212 LINES
        DATA >FFFF

* VDP REGISTERS WHEN WE EXIT
*
REG2    DATA >0000
```

```
DATA >0F00
DATA >01F0
DATA >0200
DATA >03FF
DATA >0401
DATA >0560
DATA >0E00
DATA >FFFF

END  START
```

RECENTLY RECEIVED NEWSLETTERS

Cleveland Area 99er User Groups July 93, rcvd 7/23; HOCUS May 1993 and June 1993, rcvd 7/21; TIC TOC July 1993, rcvd 7/23; K-Town 99ers July 1993, rcvd 7/12; Ozark 99er June 1993 and July 1993, rcvd 7/23 (New exchange group!)

WHAT NOT TO DO TO A DISK

by Gary Bishop

I was recently asked about two floppy disks that contained TI Writer files. It seems one of the directory sectors was blown on each disk. The cause was probably that the disk was left in the drive, and the drive door closed, while powering down the system. Try this the next time you shut your system down: remove the disks from the drives, and watch the drive select lights as you remove the power. The lights usually flash momentarily. That flash can be activating the write electronics in the drive, blipping and trashing anything that happens to be under the head at that time. Woe be to the person that does not remove the disks from the drives, or at least open the drive door, before powering down. It took some detective work, but I was able to recover most of the files from the blown disks.

FEELING FRAGMENTED?

BY SISTER PAT TAYLOR, BVM

Those of us who constantly modify files, especially graphic files, start to "FEEL FRAGMENTED!" After altering graphics, I have spent hundreds of hours in file by file copying so my disk is clean of fragmentation and will load files quickly. One of my frustrations is that so many of my commercial originals are also fragmented. Even a file by file copy won't remedy fragmentation if the disk is full and the last file is fragmented. I have tried everything, even copying the fragmented file before others trying to remedy the problem.

Now, there is a fantastic program, "THE DEFRAGMENTER" by Mark Schafer, and IT WORKS!

I dread Assembly programs, as everytime I put in the Assembly module, it takes forever to get my Super Extended Basic module to connect properly again. However, DEFRAGMENTER works from Funnelweb as well. I was so nervous trying it (Assembly does that to me), I made a back-up copy of my fragmented disk so I did not lose anything until I got the "hang" of the program.

Surprise! It did everything it promised. What is really neat too is that it can reserve space for files likely to expand or provide for those likely to decrease. An example of the problem is when I cut down a picture in TI-Artist, and had more work to do on it. I have to save it as an instance within the 22 column size for cards. That is how I fracture so many of my files trying to re-size them to fit all the programs in which I use them.

Mark Schafer only asks $6 or $7 (if he provides the disk) for this marvelous program. Its status is fairware. His address is:

539 Whitaker St.
Morehead, KY 40351

Testimonial to the above from the newsletter editor: I have a disk with many auxiliary files necessary to produce this newsletter. Every time I need to use a file from that disk, much crunching and grinding and searching goes on. I timed DSKU V4.2 at 23 seconds to catalog the disk, with 72 files, 374 sectors in use, and 30 fractured files. Mark's program took 6 minutes 36 seconds to defragment the disk, after which DSKU only required 8 seconds to catalog the disk. Now, when I ask to load a certain program, it starts loading nearly immediately, with only steady and sequential steps of the disk drive. Previously, about as much crunching was required to load a file as occurred when the disk was cataloged. A great job! I found some very slight creature features that could be added, but Mark distributed the source code with the program for your customizaton. This program is worth every penny being asked. I will bring it to the next meeting. Bring a disk if you want a copy. Remember to send Mark his money; you'll sleep easier at nights, and benefit from an extremely useful program. -Gary Bishop.

989

NEXT MEETING: Tuesday

August 10, 1993 6:30 PM

WEST MUSIC, COLLINS ROAD PLAZA, MARION

ACROSS FROM LINDALE MALL

Cedar Valley 99'er Users Group
c/o Jim Green
377 Cambridge Dr. NE
Cedar Rapids, Iowa 52402-1446

FIRST CLASS

Send To: ______________________________

GARY BISHOP
124-222
3270 28TH AVE
MARION, IA 52302