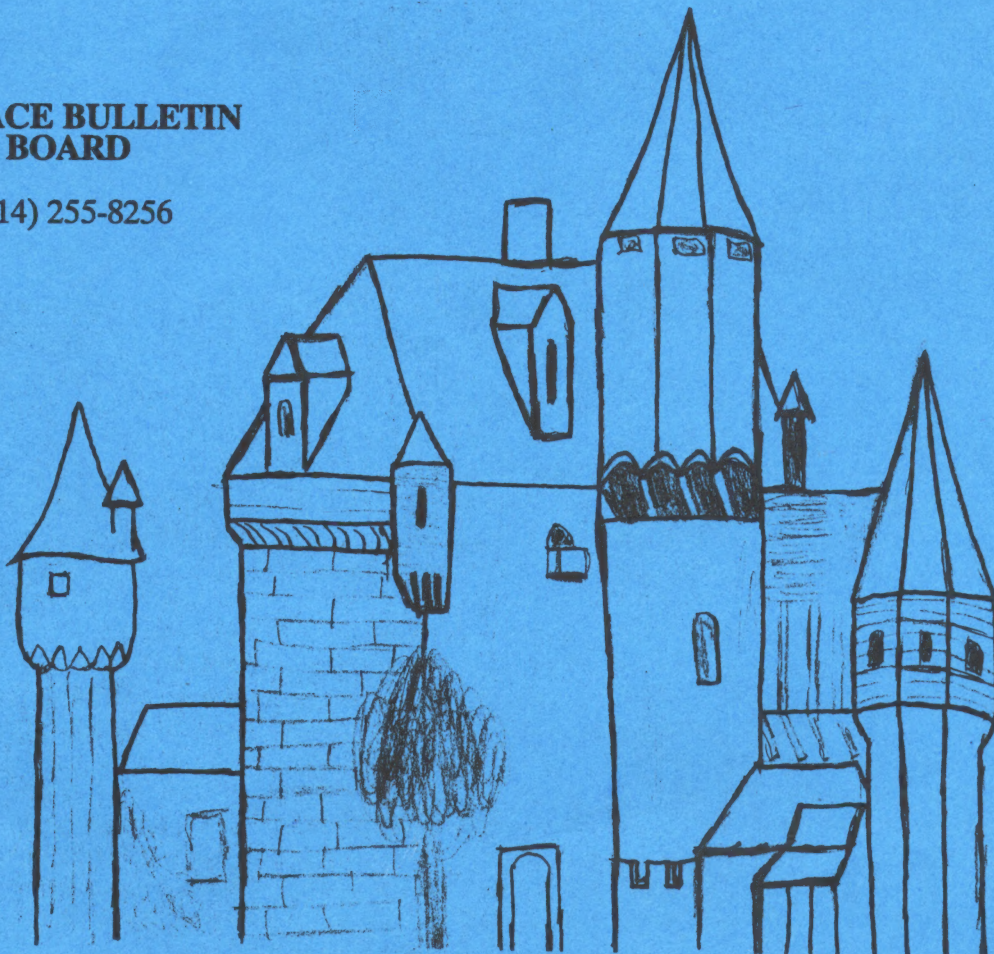


**Dallas Atari Computer Enthusiasts**

**Volume 10, Issue 8 August, 1989**

**DAL-ACE BULLETIN  
BOARD**

(214) 255-8256



**MEETING DATES: AUGUST 12, SEPTEMBER 23.**

## Highlights of the July Meeting

For those of you who were unable to attend the July meeting, here is what you missed.

### Main Meeting

The club vice president, Brenda Arnold, called the meeting to order and quickly presented the speaker for the day, Charles Marslett.

Charles is an old time DALACE member and the author of MYDOS, the first available alternate DOS for the 8-bit. As most of you are aware, MYDOS supports just about any disk drive combination available, including hard disks and true double density, and MYDOS is probably the first DOS that supported all of those drives.

MYDOS, source code and all, has been placed in the public domain. One cannot say enough about what an achievement MYDOS has been.

Anyway, as a speaker, Mr. Marslett matched his fame as a programmer. We all enjoyed an interesting interactive session as Charles gave us a run down on the latest and greatest version of MYDOS. (Charles collaborated with Bob Puff, another famous name in 8-bit circles, on this release).

We heard some ancient history, some programming insights, and some new stuff on the disk format of the SF551 drive. Remarkable session Charles, well done.

Charles has left us with a new addition to the club library, the latest release of MYDOS, version 4.51. The new version consists of three single density disks in ARCD format. (Goodness knows how many disks unarced).

Thanks again Charles for a very informative session. Also thanks to Brenda for making the arrangements for the speaker.

Treasurer Rene Tucker presented her financial report for the preceding month. There was a fair sized increase in the club bank balance thanks to the fiscal responsibility of our current crop of board members, and also to the sterling efforts of Anita Uhl who solicited and followed up on obtaining several full page ads in last month's newsletter. Good job Anita.

While we are on the subject, please let these vendors know how much you appreciate their support by buying their products. They are the only dealers we have.

Jim Jackson was elected by acclamation, to the Member-at-Large position, left vacant, when Ralph Salmeron moved out of the state. You are to be congratulated Jim, for volunteering to serve, when your club needed you.

After the usual Question and Answer session, Brenda closed the main meeting.

### ST SIG Meeting by Jim Jackson

Thanks to Megabyte in Hurst and Herb Parsons for demoing the Vidi-ST at his new store after the last meeting.

It is an exquisite piece of software for those interested in such matters. Vidi with one meg of memory will allow the user to capture twenty-five frames and save the pictures, either in Degas or Neochrome format.

This program will allow you to move (cut & paste) your items around the screen, duplicate on screen and a host of other things that this person has no knowledge of. The program is very user friendly. It is a see-to-believe piece of software.

Herb has agreed to demo this software package at the Sig meeting after our next main meeting. Stick around.

Megabyte, at 909 Melbourne, Hurst, Tx. sells this normally \$199.00 package for \$179.00 and as a special for the club will sell it for \$149.00 at our meeting.

P.S. Megabyte phone #...Metro 817-589-2950

### 8-bit SIG

The 8-bit SIG met at 10:15 am. A little late this time, so thanks to those who waited for me to show.

Club member, Sandra Higgins Hanna, demoed Dallas Quest, a program that I had never seen before, and one that has never received very much publicity. It is amazing how a gem like this can remain in obscurity. Anyway, it is a graphics/text adventure game cast at South Fork Ranch, (where else). There is Sue Ellen, and guns, and horses, and stables, and of course the famous gateway arch. We will let Sandra tell you the rest in her review later in this issue.

Dave Gramm demoed the August disk from Antic. There were some new songs for the AMP2 music program, and SUPERHOP, a very fast fractal pattern generator, and the I-CHING fortune telling program. Not demoed was an address list manager.

Antic programs are not public domain, so if you want to play I-CHING give Antic a buzz on (800) 234-7001. Ask for the August 1989 disk. The price is \$5.

Rene Tucker gave us a look at the cartridge version of DIAMOND, complete with pull down menus and ST style icons. We were able to quickly copy some files and toss them into the trash can.

The 8-bit SIG ended on that note, but continued after the main meeting with one new member getting some much needed help on using Atari Writer +, thanks to Dave Gramm.

### Dollar Signs

by Rene Tucker, treasurer

Here are the highlights of the financial report given July 8, 1989 for the month of June.

RECEIPTS	\$330.00
EXPENSES	\$160.90
GAIN	\$169.10

Remember, a full report is given at each regular meeting and copies of the monthly reports are available for inspection by any club member.

### Review of Fleet System 2

by Sandra Hanna and Harold Lewis

Sandra: Following the instructions in the appendix for Atari, I booted up DOS 2.5, then after I realized that nothing would work until I inserted the Fleet program disk and typed the RUN command. Silly me. I selected the printer from the selection given (where with Textpro, I didn't have to bother), left the color selection alone, then started looking for instructions on how to import a half completed story to work on. This proved to be a challenge. Since it was written in ASCII, I used the command to INSERT preceded by an "a". I had to do this twice for reasons unknown, but it finally worked. One thing that will be particularly frustrating to more computer literate users is the Commodore method of pressing the command keys separate and independent of each other. Until I understood this (actually, it was a suggestion from Bill, my husband. Credit where credit is due, eh what)?

Spellcheck seemed ok except that it seemed slow. I am used to a Macintosh SE using Microsoft Word, so I'm not sure if some prejudice isn't showing. It did allow me to go to the fridge for a brew and some chips while it went through the entire alphabet. When I returned (was this program driving me to drink?) the purposely misspelled words were highlighted and the first one was flashing, waiting for my more intelligent input. Verdict: Slow but perfectly satisfactory. The feature that's nice for me as a writer is that I can add colloquialisms and computer-eze to the dictionary so that they won't come up as "misspelled" afterward.

Deciding to create a paragraph where I began to talk about the Spellcheck feature, I-as with virtually every computer I have worked with before-placed my cursor to the left of the "The" at the beginning of my sentence and hit "RETURN". Bad mistake. It deleted that first line. No matter where I looked in the instructions, did it mention such a move would produce this effect. I tried it again, thinking I must have pressed FUNCTION-something and simply forgot. (Curse word deleted)-----if it didn't do it again! I decide on having another cold one on the patio (its 102 out there), shut the thing off. Ah he who fights and runs away...

Still working on the paragraph problem, I return, days later, read more of the manual, and decide to try to "Insert a line". I pressed the "Function" key, then held the "shift" as I pressed "insert". Instead of a blank line being inserted into the space between the ? and the next word, a blank line was inserted ABOVE my line. I tried again. This time, my line, including the ? was deleted. Grrrr. I begin to look up "Insert Text". I find "Transfer Text" hoping it will retrieve my text

from the ether. It didn't. I ended up typing it again. I am not happy with this program. Maybe with practice, I could become comfortable with using it, but I still feel Textpro is a superior word processing program. I do wish, however that it had the Spellcheck, and in The Fleet 3, the Thesaurus feature. I don't like the way in which you have to insert space after you delete copy and how it automatically erases things without warning. I'm happy with Textpro. This program would be fine for high school or college students to do reports, or for the business man to type letters, but there are better and faster programs available, and I have a Thesaurus and Spelling Dictionary that sits by my computer at all times. No thanks. Fleet 2 is anything but.

Harold: I thought Fleet 2 was a pretty good program, though, like Sandra, I wasn't real impressed with the fact of having to load DOS 2.5 to get into the program. Also the instruction book was hard to understand because of flipping around from Tutorial to the special Atari section. The program was made to work on a Commodore, and then the Atari. Its section was in the back of the book. I've tried Atari Writer (cartridge) and this program is better. I've heard Atari Writer now has a Spellcheck feature but I haven't tried it.

I use Fleet 2 for letters and for banking records. Its great for that. I agree with Sandra that it would be good for students. Sandra's main problem, I think is that she is comfortable with Textpro. We both ask the question, "Has anybody heard of Textpro with a Spellcheck?"

I'm working on learning Textpro now, so I'll be able to compare soon. Maybe Sandra will win me over, but for now, I'll stick with Fleet 2.

## Touch Up

### An ST review

by Angela Burns

A few months ago, I was putting the final touches on the Dal-ACE newsletter when I decided to change an .IMG picture file to make it fit in a little better with the ideas on the page. I tore through the .IMG Scan, Image Pro, and everything else in my library that I thought could possibly do the job; nothing did. (Of course, the aforementioned programs are great at what they do; they just don't do what I was looking for.) About two weeks later, I heard about Migraph's Touch Up; that was one program I really got excited about!

When I finally got Touch Up in my hot little hands, I found the experience a strange mixture of euphoria and disappointment. Touch Up does everything it claims it will do; that's a nice change from some programs I've bought. But I found some drawbacks that make me wish there were an alternative.

#### Price

Touch Up retails for \$179.95. That's pretty steep for an ST product. Yes, it's a bargain compared to many programs for other computers (notably some IBM retail monstrosities), but it's still a chunk of change. I think the price will limit the already limited ST market to those users who absolutely have to have it for their design work. Users who would like to have it but don't absolutely need it will probably settle for the shareware image editor on the BBSes (which is very limited when compared to Touch Up) or wait and pray for someone to market a similar program at a more affordable price. So that's drawback number one.

#### System requirements

Touch Up will run on any ST with at least one meg and a double-sided drive. It runs in either color or monochrome. It also supports hard drive usage.

Migraph says that you may not be able to load some very large images with only a one-meg machine, but I wouldn't let that worry me. I haven't run into any file size related problems yet, and I've worked on images of over 100K. Yes, I know that you can produce images much larger than that when using, for instance, the .IMG Scan at high resolutions; but I feel that, for most purposes, fairly small images scanned at 150-200 dpi are more than adequate. Most desktop publishers have little practical use for magazine-quality resolution.

#### Copy protection

The Touch Up disk is unprotected, so you can make backups with ease. That is not to say that the program isn't protected. It comes with a hardware Program Security Key (PSK), known in the pirate community (for some reason) as a dongle. (Ed. note: The August issue of Computer Shopper reports that the latest version of Touchup does not need the PSK, upgrades are available, and there is a \$15 credit for the return of the PSK. Mygraph 800-223-3729).

The PSK is installed by plugging it into the parallel port of the computer. You then plug your printer cable into the PSK. Touch Up checks for the presence of the PSK and will not run if it's not there. The PSK will add about two inches to the space needed behind your ST.

I understand the reasons that companies continually try to protect their software, and I have a lot of sympathy for them. They work very hard to get a product to market, and they deserve to be paid for their efforts. It must be agonizing

to find your program distributed on the pirate boards. Still, copy protection of any kind doesn't work. I was told by a very reliable source that Touch Up was on the pirate boards within a couple of days of its release; the cracked version worked perfectly. So, the PSK (which retails for around \$25.00 by itself) only accomplished one thing: it made Touch Up more expensive and cumbersome for the legitimate user. Drawback number two.

#### Documentation

The manual is in a very nice boxed notebook form. There are seventeen chapters with lots of illustrations. It's not the sort of thing you can just sit back and read; Migraph suggests (and so do I) that you walk through the program and try out the features as you read about them.

The table of contents is detailed enough that the absence of an index is only a minor drawback, but I still feel there should be one. It would be a little easier to find things with a detailed index, and Touch Up requires numerous trips to the manual to look things up.

#### Features

There are no drawbacks in what Touch Up does. There are so many features, it's hard to tell where to begin describing them!

Touch Up is an editing tool for monochrome images. You can load color images, but you must change them to black and white before you can work on them. This goes for color monitors, too. Mapping a color image to monochrome is done in one of several ways. Simple mapping is the default and is recommended for images containing few colors. Using simple mapping, you can choose to have ALL colored areas changed to black, resulting in a stark image with little detail;

MOST colors mapped to black, resulting in colors that are about 30% black turned to solid black and the rest changed to white; or SOME colors mapped to black, resulting in only dark colors (those that are about 70% black) mapped to black. If simple mapping doesn't give you the image you're looking for, try one of the two algorithms Touch Up uses. The F-S (Floyd-Steinberg) and Burke algorithms change different colors to different textures, rather than changing everything to simple black or white. In addition, you can have the image auto mapped to a 2x2 or 4x4 grid, producing more detail. (In other words, you can change a 320x200 image to 640x400 or 1280x800 using this feature.) When using a color system, you can use the table mapping feature, which is the same as auto mapping except that you control the patterns yourself.

The images you can produce with the program are bit images. That is, one pixel on the screen corresponds to one dot in the image. The images are not limited to screen size but only by the amount of memory you have (except for Degas and MacPaint pictures; those are limited to the size of the screen.). And, as I said, it takes one heckuva file to be too large.

Touch Up can load files in any of the following formats: .IMG, Degas, Degas Elite, MacPaint, NeoChrome, .PCX, .GEM, and .TIF. Save files in those formats and in IFF-ILBM and PrintMaster formats. You can also save a clip area of an image or load an image into a clip area to combine more than one picture.

There are several "modes" which provide different tools for creating and editing images. In each one, you may use different writing modes: replace, transparent, reverse transparent, and XOR. These choices provide you with

practically unlimited effects. For instance, by filling an area with a pattern, then filling it again with another pattern in a mode other than replace, you can create many custom patterns. Other choices, such as line styles, fill patterns, and shadow (to create a shadow effect for the objects you draw) also apply to several modes. In all modes except Lightning (which works only at normal size), you can change the screen display to show the full page, normal size, or magnify the image by 2 or 4 times. Fill patterns can be set to match the resolution of your printer; i.e. if you will be using a 300 dpi laser, use 300 dpi fill patterns.

Most commands can be accessed in two ways. There is an icon pad at the left of the work area where you can click on the tools you wish to use; the tools on the pad change when you change modes. There are also drop-down menus. Some commands also have "quick keys", but it's easy to forget them, since they are not stressed in the manual and there's no on screen reminder of them (like in Word Writer ST; I like those). The icon pad can be made into a "tear-away" menu; it is then a window that can be moved around the work area. This is mostly useful for those who have large-screen monitors.

Drawing Mode provides most of the features you usually associate with a paint program, such as polylines, boxes, circles and ellipses, and fill. You can use this mode to resize or clear the page, as well. This is where you set the default drive paths for fonts and for saving your images. You can use four different sizes of pen for free-hand drawing. A really neat feature is the B-Spline and Bezier curves. Use these to create very precisely curved lines. Bezier curves have four control points that you can move to create just the curve you want before placing it

down on the page. B-Spline curves can have up to 32 control points! Much easier than using the free-hand drawing tool in a standard paint program to try to draw complex curves with a mouse!

Clip mode allows you to "clip" out an area of the image to modify only that area with special effects. You can stretch, copy, move, or slant a clip area. You can set the color of a clip area to black or to white, or invert its colors. You can cut out a clip and save it to a temporary clipboard file on disk for later use, then paste it back when you're ready for it. Use Save Clip Area to cut out the part of the image you wish to save and cut out all the useless white space around it. You can also flip, mirror, slant, or rotate (in 90 degree increments only) a clip box. And you can clean up stray black or white dots in a clip area. Use Outline to convert solid black areas into white areas outlined in black (good for when you want to fill an area with a pattern other than black). Use Mask to overlay a new pattern on an area. Also in Clip mode is the IMG Viewer; use it to get a preview not only of the image itself but of information about the file, such as the file size.

In Text mode, you can use a variety of type styles to add text to your images. You may give the type various attributes, such as bold, outline, or filled, and use any pattern to fill it. The height of your text is measured not in points but in pixels. At first, I thought this was pretty bizarre. Later, though, I realized that this was the best way to do text. Since you sometimes don't know at exactly what size the image will be printed, what you really need to measure is the size of the text relative to the rest of the picture - in pixels. After you have placed the text on the page, it is a part of the image and you can use

clip mode, etc. to create different effects.

Most of your editing work will probably be done in Lightning Mode. This is a full-featured paint program within the larger program. In Lightning mode, you may only work on the screen area. Moving to a different area of the image requires that you leave lighting mode, scroll to the area you wish to work on, and return to Lightning mode. (I do feel that it should be possible to move around the graphic without having to do this.) It is faster than Draw mode, and it is the only part of Touch Up that offers an UNDO feature. You can undo either the last operation performed or everything you did since the last time you entered Lightning mode. In Lightning mode, you may use any shape (such as a box or circle you have created) as a paintbrush. You may also "lasso" any piece of the onscreen image, then use that piece as a brush. You can use a fill pattern as the "paint" in your spray tool, and you can set the spray to "flow" at different speeds with a round or square brush of three different sizes. You still have all your boxes, ellipses, etc, and the B-Spline and Bezier curves. The Fill feature only works out to the edges of the screen; for filling areas larger than the screen, you must use the fill in Drawing mode. You can also fill black areas; something that isn't possible in some paint programs. Also offered in Lightning mode is FatBits, a zoom feature you can use to edit detailed areas, one pixel at a time. Lightning mode has a great eraser; use the mouse to size and shape it to suit your purposes. The eraser can be as large as the entire screen or very tiny.

File mode is where you load and save images, load files into the clip area, and print out images. (Only users who have at least two meg of memory can print from within

Touch Up.) You can also access the .IMG Viewer from here.

Scan mode is currently inoperative. A future version of Touch Up will work with scanners so that you can use it to scan with as well as edit. Registered users will be notified when that version is available.

Other stuff: Color buttons let you set the ink color to black or white quickly. The Locator is a small display that shows you how much and what part of the screen you are currently working on. You can grab the black square that represents your work area and move it to the part of the page you want, rather than having to scroll around.

You can exit Touch Up directly to the Outprint program if you wish to print out your images. Outprint evidently works just like it does in Easy Draw; to be honest, I haven't tried it. I always print my images out from Timeworks DTP, since what I want to know is how they print out from there. I have only had one file (out of a couple of hundred) that has refused to be printed from DTP after it was saved from Touch Up, and I can't be sure exactly why it didn't work. Anyway, you can send the images to the printer or print them to the screen, like a slide show. And you can queue up files to be printed one after another, instead of having to load one, print one, load one ... etc.

#### The rest of the story

MACCELL.PRG is a mouse accelerator program from Atari; Mi-graph has included it on the Touch Up disk. Why? Because the way this program responds to mouse clicks is so slow as to be unbelievable. For a while, I thought my mouse was dying. I used the program for a few days without the accelerator, then for a few days with it. Eventually, I stopped using MACCELL.PRG because it seems to me that it just gives you the illusion that you are accomplishing

things faster because the pointer whizzes around the screen so fast. The program still takes forever to respond to mouse clicks, even when selecting files or other operations having little to do with the program.

The user interface is the biggest drawback of all. It is confusing. Some commands are executed with a left mouse click; some with a right click; some with a left, then a right; some with a double right click. Double right click? Yes, believe it or not. You cannot make the best use of this program without constant reference to the manual. A consistent user interface would make Touch Up great; as it is, it is merely all we have right now that will do the job.

#### My recommendation

If you have a real need for a program that will edit .IMG files for your own important design work, then by all means, buy it. It does the job and will give you outstanding results, though not without a certain cost in terms of being a pain to work with. Otherwise, you may be better off just buying clip art off the shelf, trying out the far less expensive shareware image editor accessory, or waiting for someone else to write an easier to use, less expensive scanned art editor.

---

#### Editor's Note of Appreciation

Many thanks to Angela, Sandra, Harold, Bill, Jim, Rene, Dave, John, Tony, and especially Anita, for helping me to fill the pages of this newsletter. Thanks to you all, the newsletter is looking pretty good, and I might even be able to meet the publishing deadline. You all come back again next month, yo' hear.

---

## Dallas Quest

### 8-bit Review

by Sandra Higgins Hanna

For all of you who missed the demonstration, or just failed to take notes, I hope that this will entice you to buy the game. The graphics are first rate, the music is by one of Hollywood's finest, and the game is witty and challenging; so challenging in fact, that you may die of old age trying to best it. Ah, but you'll die with a smile on your face.

After inserting the disk, a graphic representation of the famous characters from the "Dallas" TV show appear in the upper half of your screen. Then, following the instructions to flip it over and reinsert, the scene changes to the front gate to Southfork as the credits scroll beneath, complete with the whole score faithfully reproduced in computer music. The song ends and the scene changes to the living room with Sue Ellen lounging seductively on the couch. There is a bar above the graphic that lets you know how many objects you are carrying, which direction choices you have, and how many clues there are.

So, back to Sue Ellen, right? Right! The text guide prompts you saying "There are three things in the room, Bugle, rifle, Sue Ellen. Pretty hard to miss her, huh, guys. The activating words are typical to this type of game, GO, GET, LOOK, DROP, etc. So you get bold and...nah, I won't spoil your fun. What I will divulge is that I haven't yet made it out alive, which seems to be the object of the game. (I have lost my Tutorial of Dallas Quest so I'm guessing here.)

I suggest keeping a log of what happens when and maybe even draw a map of the house. The house layout is always the same, but different things happen de-

pending on what you look at, get, or do. There are several ways to die, all of which have great animated graphics to match the death. Buy it, you'll like it. It's fun and I've never ceased to have an interesting session with the game each time I play it. It is not an ego booster, though, so if you've been recently laid off, or received your 21st rejection letter for your latest manuscript, wait a while. It can be frustrating and I'd hate to have you take it out on the wife or the dog...or me!

So, I bid you farewell and riding off into the sunset, I shout over my shoulder, "Enjoy!"

## ASK TIM

by Tim Sharpe

Michigan Atari Magazine

Reprinted from ACORN, June 1989 issue

Q. Why is the 810 disk drive so large?

A. This is a mysterious phenomenon indeed. To find the answer, I contacted a few Atari insiders and they told me the shocking truth; the 810 was originally designed to run on diesel fuel! Atari's reliable old clunker was supposed to have an engine under the hood; but it seems the 810 couldn't meet the government's minimum fuel consumption requirements for disk drives (10,000 sectors per gallon), and there was also a problem with the exhaust system. It's a pity the "Diesel 810" didn't work out - it probably would have run quieter.

Q. Are the sixteen-bit Ataris better than the eight-bit Ataris, or vice versa?

A. This has indeed been a controversial issue. I say "been" because I have conducted a fair and impartial test of my own and have found the answer. Since I own both an Atari 800 and a 520ST, it

was quite easy to determine which was "better". I gave each machine three tests and have listed each and the corresponding result below:

The Water Test: It is a well known fact that bad computers sink when immersed in water, and likewise good computers float. I filled a bathtub with water and placed both the 800 and the 520ST in it.

Results: The ST went straight to the bottom. The 800 was buoyant beyond belief, and the cartridge hatch provided ample room for a rubber duckie. Chalk on up for the eight-bits.

The Punt Test: An important feature of any computer is its ability to be kicked long distances. This action is necessary after running a difficult program or encountering a zillion bugs in a program you wrote and thought would work the first time. It relieves tension and lets you live a longer life.

Results: Both computers weighed roughly the same and could be kicked with about the same force. However, since the ST is built much like an airplane wing, it flew three times the distance the 800 did. (The mouse acted like the tail of a kite and steadied the ST in flight.) Chalk one up for the sixteen-bits.

The 100 Meter Freefall: Crucial to any computer is its ability to be dropped. Accidents happen.

Results: After finding a particularly nasty precipice, I dropped both computers simultaneously. Both shattered spectacularly. This one is a tie.

I have come to the conclusion that both machines have their own unique advantages and disadvantages, and that either machine is preferable to the typewriter I'm typing this on.

## PRESS RELEASES

The following is a sampling of the press releases recently received by the club. These products have not been tested, and as such, carry no recommendations, pro or con, from DAL-Ace. For more information on these products, consult the press releases at the club meeting.

### The A Exchange

A user's newsletter for all Atari owners. Features: Gamers Corner and Marketplace (buy sell trade advertisements). Subscription rate: \$12 per year, 12 issues. P.O. Box 25852, Fresno, Calif. 93729.

### Spectrum Holobyte's Tetris

Tetris has received four awards from the Software Publishers Association. Best Entertainment, Best Action, Best Original Game, and Critics Choice. Similar to the Red Squares game we demoed at the last meeting only with enhanced music and graphics. See review in last month's newsletter.

### Regent Word, Student Edition

\$25 GEM word processor with 40,000 word spelling checker. Double column printing, unlimited printer support, and no copy protection. Will run on a stock 520ST.

### Abacus BeckerCAD ST

Professional Computer Aided Design software for the ST with at least 1 meg of RAM, (2 meg recommended). Supports Postscript printers including the Atari Laser with Ultrascript, and some dot-matrix printers. \$395.

### Migraph Hand Scanner

Scans a strip 4" wide, resolution 100-400 dots per inch. Bundled with Touch-Up software. Images can be saved in a variety of formats,

IMG, PCX, TIF, Degas. Retail \$499, Special offer \$429. (800) 223-3729.

### WinterTech Solar System

The computer guide to the solar system, 3-D graphics and animations, and much much more. Introductory offer \$44.95 ST (407) 425-1199

### Mindscape's Thunder Blade

Sega coin-op hit. Attack helicopter simulation. \$50 ST

### Mindscape's Action Fighter

You are the Super Rider. Start out on a motorcycle, find the right parts and change it to a super-charged car or an aircraft. Ground and air battles. \$40 ST

### Mindscape's Road Raider

Action packed road adventure by author of Infiltrator. Rave reviews of graphics and playability. \$50 ST

### Mindscape's Captain Blood

A dazzling trip through the Galaxy. \$50 ST.

### SSI's Demon's Winter

Fantasy role playing game. Just released.

### SSI's Red Lightning

World War III war game. Release date July.

### SSI's Pool of Radiance

Fantasy role playing game. To be released in future.

### SSI's Star Command

Science fiction game. To be released in future.

The following titles were previously announced, but the decision has been made to NOT release them for the ST:

Panzer Strike!  
Sons of Liberty

## The 130XE/320K Upgrade by Scott Peterson

After both reading and building both the 800/288K upgrade (D.G. Byrd), and the 800XL/256K upgrade (C. Buchholz), I decided that there also had to be a way to upgrade the 130XE. There is, and thanks to the "Freddie" chip (CO61991) this modification is much easier to do than either of the other upgrades.

To do the upgrade you will need a soldering iron, de-soldering tool, and some fine wire. See the parts list for the chips needed.

First, remove both the case and the metal shield to get down to the mother-board. Then remove the eight ram-chip U26 thru U33 (MT4264). They are the row closest to the TV RF module. Next, install Z2 thru Z9 in the place of U23 thru U33. These are the 256K ram-chips. You can solder them to the mother board, or use sockets. Now take a piece of wire approx. 12 in. long and run a jumper from pin one on each of the 256K ram-chips to the next. After you do this the wire will be connected to pin 1 on Z2 thru Z9 and you should have about 6 inches left over. Do this on the rear of the mother board and then snake the wire thru the large hole near the ram chips.

Next, desolder and remove U23 (CO14795), and replace it with a 40 pin socket. Bend up pins 15 and 16 on U23 and insert it in the socket you just installed. Take Z1 (74LS158) and bend up all the pins on it except pins 8 and 16. Put this "piggy-back" on top of U20 (HD14050) and solder pins 8 and 16 of Z1 to pins 8 and 16 on U20. Now solder a short jumper from pin 15 on Z1 to pin 8 of Z1.

Now, take a piece of wire about 4 in. long solder one end to pin 30 on the chip marked "CO14805" on the mother board, and the other to pin



1 on Z1. Next solder a wire to pin 15 (one of the two you bent out) of U23 and connect the other end to pin 2 on Z1. Solder a wire to pin 16 on U23 and connect the other end to pin 3 on Z1.

Take R1 (33 ohm) and trim the leads to about 1/4 in.. Take the wire you connected to pin one on the 256K ram-chips and solder it to one end of R1, solder the other end of R1 to pin 4 on Z1. Re-assemble the RF shield and case and you are finished.

**PARTS LIST**

- Z1 74LS158 (2 to 1 Multiplexer)
- Z2-Z9 41256 dynamic RAM (150ns)
- R1 33 ohm 1/4 watt resistor
- 1 40 pin socket
- 8 16 pin sockets (optional)

The next page is a quick overview of the bit table and numbers to be used in location 54017 (PORT B). I have finished modifying a ramdisk handler for the extra ram. It uses a ram based OS so basic XE or XL can't be used. At present the best deal for this mod. is to use MYDOS 4.0. This supports a very large single density ramdisk. With basic XE you can use a 1500 sector ramdisk and without it you can have about 2000 sectors.

This upgrade has been built and tested on a BBS, it has run for days on end without a memory loss or error. If you need help or more information feel free to call the Peanut Gallery (408)-384-3906. 24 HR, 300/1200 baud. Leave mail to Sysop (that's me). Good luck and let me know if you write a better handler.

**130XE/320K Memory Control Register - 54017 (\$D301)**

Bit 7 6 5 4 3 2 1 0  
 D a b C c d B R  
 D=0 enable diagnostic ROM.  
 B=0 enable BASIC ROM.  
 R=1 enable OS ROM.  
 C=0 enable extended RAM.  
 abcd= memory control bits.

Bank #	Control #
Bank 0	>131:
Bank 1	>135:
Bank 2	>139:
Bank 3	>143:
Bank 4	>163: Basic = off
Bank 5	>167: OS = on
Bank 6	>171: ENH = on
Bank 7	>175:
Bank 8	>195:
Bank 9	>199:
Bank 10	>203:
Bank 11	>207:
Bank 12	>227:\
Bank 13	>231:130XE\
Bank 14	>235:Banks/
Bank 15	>239:/

If you are using MYDOS 3.016 and wish to use Basic XE and a ram-disk at the same time, boot DOS and poke 5275,163 and 5324,16. Go to DOS and write the new DOS. This will keep the two from "bumping" into each other.

A similar poke can be done to DOS 2.5, it is poke 4838,163. The handler I have will set up 192K of the extra ram as 2 SD ramdisks or 1 DD ramdisk.

If you are a hot-shot programmer (I'm not) I think a print spooler that uses part of this ram would also be very nice. This mod is easy to do and perfect for running a BBS. One note, on CompuServe there is a mod by Rich Andrews which should not be confused with this one, his uses 33 new chips and mine uses 9 new chips. Have fun.

**Irving World of Atari??**

Computer Shopper now confirms the report that the World of Atari show is to be held in Irving in August. We wish we had more information at press time, but we don't.





Computer Council of Dallas, Inc.  
1950 Stemmons Freeway, Box 277, Dallas, Texas 75207

**PUBLICITY  
RELEASE**

Number 0 -- August 1989. By: The Journal group.

## ACCESS GRANTED!

This is prelude to a series of columns dealing mostly with the Computer Council of Dallas. It is written by folks in the CCD-specifically, the "Journal" wing of the CCD Publicity committee.

Our intention is to provide your group's newsletter with a monthly article. We've asked all the groups with newsletters to participate. We have several good reasons for making a monthly column contribution.

We have a need for communication within the CCD membership (that's you). Our aim is to get help from you that will improve Council operation.

Second, there seems to be enough interest to support a regular column on the Council and all the groups involved. We've begun to discover that our community of users is really interested in the CCD, from its past to its future plans. You want to hear why we do things certain ways, to know how it all works. This column will give you a chance to see. It should supplement any reporting you get from your regular council representatives.

The third reason is that we can clear up some misconceptions newcomers may have about the CCD, its relation to the groups, Infomart etc. While we're at it, our articles can tackle generally user-group related subjects.

Four, we think we can add in something Council affiliation has always promised: better inter-group communications. All of us meet in the same place on the same day. We want to promote a sense of unity of purpose among all users.

So now you know: we have high goals, and this column should help us to achieve them. Before the CCD can grow further, more people will be directly involved and aware of its activities.

The column is called "Access:" because you'll get to hear what the CCD truly is all about. If you want to know more about our projects, we invite you to ask us. We're happy to share information (that's our job). We want you to share in this communication resource.

So that's what this column is doing-- we're making you a part of the inner process we've begun. "Access:" will explore and explain the who, what, and why Behind the CCD and its community of users.

(Any feedback, comments or suggestions should be forwarded to:  
CCD Publicity 1950 Stemmons Freeway Box 277 Dallas TX 75207. )

(c)1989 CCD Publicity Committee

# FORTH TUTORIAL by Jeff Golden

## Lesson # 3 Arithmetic

Forth normally uses 16-bit binary values with the hi-order bit serving as the sign bit. This form of arithmetic is called integer math, and division will produce an integer quotient and an integer remainder. Standard Forth can also use 32-bit, (double word), signed binary values for greater precision within integer math. The 32-bit binary math will handle up to 9 decimal digits with sign.

Forth is not limited to integer math, since anything we want can be added to the language. Some commercial Forths do provide a non-standard floating point package, and there have been public domain floating point packages written and circulated. Floating point may be great for the scientific world, where close is good enough, but it is not so great in the commercial world where we need to keep track of our remaining pennies.

8-Bit. Be sure you are in DECIMAL mode before trying any of the following examples. For the duration of this lesson, we also need to make a temporary patch to .S so that it responds to values above 32767 in the same manner as almost all other Forths. 8-bit enter the following:

```
DECIMAL 12705 10335 ! <cr>
65535 .S <cr>
      ( should print 65535 )
XXX   ( clear stack )
```

All. Referring to the reference chart, it tells us, through stack notation, that to perform a 16-bit ADD we need two input values on the stack and that the + function will leave behind the sum as one 16-bit value. Try the following:

```
2 2 + .S <cr>
      ( two plus two equals four )
2 + .S <cr>
      ( four plus two equals six )
7 - . <cr>
      ( six minus seven equals -1 )
```

Note that we used the dot print function on the last statement, because .S would not have recognized that the value was negative since .S assumes unsigned values. (as modified on the 8-bit).

Note also that the seven which was on the top of the stack was the subtrahend.

```
.S <cr>      ( stack empty )
```

The dot-print function ate the -1 from the stack leaving the stack empty which is normal for most Forth words except .S. We need to take a hard look at what a -1 looks like in binary arithmetic, so put the -1 back on the stack:

```
-1 .S <cr>
      ( 65535?? unsigned value of -1 )
2 BASE ! .S <cr>
      ( 1111111111111111 binary -1 )
HEX .S <cr>
      ( FFFF hex value of -1 )
DECIMAL . <cr>
      ( -1 decimal value )
```

Earlier in the tutorial, we tap-danced around the subject of hexadecimal, twos-compliment binary arithmetic, etc. by suggesting that you acquire that knowledge elsewhere. In examining my available reference material, I found that the other authors of even assembler manuals were doing the same.

"Mapping the Atari" does attempt a feeble explanation, but we will continue to tap-dance. If you do not already know binary arithmetic, we suggest grabbing someone who understands the subject for a long one-on-one session. It is not an easy subject to adequately cover in a short written tutorial. Having a full understanding of binary arithmetic is not essential to performing math with Forth. So if the subject is fuzzy to you, kind of pick up what you can, and move on. 2 2 + still equals 4.

Anyway, the above example should indicate that a -1 will have all 16 bits turned on, and if you were to add up the decimal value of 16 bits you would find that they equal 65,535 which is where the 65535 unsigned readout comes from. The left-most bit is the sign-bit and a 1 in that bit actually represents a negative value. Try the above example again only this time use -2. Then try the maximum negative number of -32768. (make sure you are in decimal mode). Observe the effect of the three different negative numbers on the bit settings as shown in the binary display. If you want to, you can try some positive numbers.

The maximum positive number in 16-bit signed arithmetic is 32767. Using a higher value will flip the sign bit on. We can use values up to 65535 when the function calls for 16-bit unsigned parameters. This is accomplished by assuming a positive value and using the sign bit as a meaningful part of the value.

Since the parameter stack is only 2-bytes (16-bits) wide, a 32-bit value is represented by two 16-bit values placed next to each other on the stack with the most-significant 16-bits placed on the top of the stack. The high-order bit of the 32 bits is the sign bit. To convert a 16-bit value to a positive 32-bit value simply place a 0 on top of the 16-bit value being converted. Enter the following:

```
65535 <cr>
      ( max value for 16-bit unsigned )
0 .S <cr>
      ( convert to a 32-bit value )
D. <cr>
      ( print 32-bit value using D. )
65535 0 <cr>
      ( reenter the double word )
1 0 .S <cr>
      ( enter a double word = to 1 )
D+ .S <cr>
      ( do a double add and chk stk )
D. <cr>
      ( print value of 65535 plus 1 )
```

The above exercise should give you an idea of how adding one to the max value that can be held in 16 bits will cause the 17th bit to turn on. Note that the most significant 16-bits are on the top of the stack so the progression to the 17th bit which is on the right may appear to be a little strange.

To convert a 16-bit negative value to a negative 32-bit value, place -1 on top of the negative 16-bit value. i.e. entering -1596 -1 <cr> would place a 32-bit value on the stack equal to -1596. Entering D. <cr> will print -1596.

Up to this point, we have been demonstrating the fact that a 32-bit number is nothing more than two 16-bit values placed side by side on the stack. It is important to know how to convert 16-bit values to 32-bit values. Forth does provide an easier way to put a 32-bit value on the stack other than the method we have been using.

Entering a value that contains a decimal point will automatically create a 32-bit value on the stack. Try the following:

```
65536. .S D. <cr>
655.36 .S D. <cr>
```

At this point in time you should have an understanding of the basic operation of 16-bit and 32-bit arithmetic. You should also have an understanding of stack notation and should be able to read and understand the following descriptions of the available arithmetic functions. So we will leave you to peruse the definitions. Devise your own simple arithmetic problems and experiment with the functions.

### Arithmetic Word Definitions:

**+** ( w1 w2 -- w3 ) Pronounced "plus". Adds the 16-bit signed values w1 and w2 and leaves the sum w3 on the stack.

**-** ( w1 w2 -- w3 ) Pronounced "minus". Subtract w2 from w1 leaving the result w3 on the stack. All values are 16-bits with sign.

**D+** ( wd1 wd2 -- wd3 ) Pronounced "d-plus". Adds the 32-bit double word signed values wd1 and wd2 and leaves the double word signed value wd3 on the stack.

**D-** ( wd1 wd2 -- wd3 ) Subtracts the 32-bit double word signed value wd2 from wd1 and leaves the result wd3 on the stack. Note, D- is a non-standard word that is present on the ST system only. You can add D- to the 8-bit system with the following code : D- DMINUS D+ ; <cr>

1+ ( w -- w+1 ) increment w by one.

1- ( w -- w-1 ) decrement w by one.

2+ ( w -- w+2 ) increment w by two.

2- ( w -- w-2 ) decrement w by two.

\* ( w1 w2 -- w3 ) Multiply w1 by w2

leaving the result w3 on the stack.

**UM\*** ( u1 u2 -- ud ) unsigned multiply giving double precision unsigned result.

**UM/MOD** ( ud u -- mod quot ) unsigned division using double precision dividend and single precision divisor and results.

**2/** ( n -- n/2 ) Arithmetic right shift (divide by 2).

**/** ( n1 n2 -- quot ) divide n1 by n2. All values are 16-bit signed. See /MOD for more information on negative quotients.

**/MOD** ( n1 n2 -- mod quot ) divide n1 by n2 giving remainder and quotient. All values are 16-bit signed. As long as you are dealing with positive dividend and divisors, the results on either system are what you would expect. i.e. +5 divided by +2 gives a +1 remainder (modulus), and a +2 quotient. On the fig-Forth system, all quotients are rounded towards zero. For some strange reason, not understood by this author, the framers of the Forth-83 standard, after a heated debate, decided that negative quotients should be rounded toward negative infinity. At any rate, if you must deal with negative dividends or divisors, I would recommend experimenting with some simple divides to ascertain just what the rounding and sign effects will be for the system you are using.

**MOD** ( n1 n2 -- n3 ) divide n1 by n2 leaving the signed modulus (remainder).

**\*/MOD** ( n1 n2 n3 -- mod quot ) Multiply n1 \* n2, and then divide by n3 using a double precision intermediate step.

**\*/** ( n1 n2 n3 -- quot ) Same as \*/MOD except gives quotient only. \*/ is good for handy approximations with a precision better than 10 to the 7th. pi as 355/113; e as 25946/9545; square root of 2 as 27720/19601; square root of 3 as 32592/18817; and square root of 10 as 27379/8658.

**MAX** ( n1 n2 -- n3 ) leave the larger of n1 or n2 on the stack.

**MIN** ( n1 n2 -- n3 ) leave the smaller of n1 or n2 on the stack.

**ABS** ( n -- u ) If n is negative, make it positive.

**NEGATE** ( n1 -- n2 ) ( use MINUS in fig-Forth ) If n1 is positive make it negative. If n1 is negative, make it positive.

**DNEGATE** ( d1 -- d2 ) ( use DMINUS in fig-forth ) Same as negate only values are double words.

So much for arithmetic. If you want to, you can bring up your systems and play around with the various arithmetic words for the purpose of viewing the effect on the stack. For the most part, these words should be self explanatory.

## Lesson # 4 Memory and strings

**@** ( addr -- x ) Pronounced "fetch". Get the 16-bit value at the memory address and place it on the stack.

**C@** ( addr -- byte ) Pronounced "c-fetch" Character fetch. Get the 8-bit value at the memory address, convert it to a 16-bit value by appending hi-order 0s, and place it on the stack.

**!** ( x addr -- ) Pronounced "store". Store the 16-bit value x at the memory location addr.

**C!** ( byte addr -- ) Pronounced "c-store". Character store. Store the least significant 8-bits of the stack item at memory location addr.

**+!** ( w addr -- ) Pronounced "plus-store". Add w to the 16-bit value at the memory location addr.

**CMOVE** ( from to u -- ) Move u bytes starting at from-addr to to-addr. This is a fast machine language move, and movement progresses from left to right. If operands overlap, CMOVE can be used to slide a string to a lower address.

**CMOVE>** ( from to u -- ) Like CMOVE except that movement progresses from right to left. If operands overlap, CMOVE> can be used to slide a string to a higher address.

**FILL** ( addr u byte -- ) Fill u bytes of memory at location addr with byte. Can be used to clear an area of memory.

**COUNT** ( addr1 -- addr1+1 byte ) Move string count from location addr to the stack in preparation for TYPE. Strings will often contain a one byte count at the start of the string. This word moves the count byte to the stack and updates the address so that it points to the beginning of the actual string characters.

**-TRAILING** ( addr n1 -- addr n2 ) Search the tail end of the string at addr and modify string count n1 to eliminate any trailing spaces. Usually used before executing TYPE.

# PROGRAMMER'S CARD

Hardware Register				OS Shadow			
Name	Description	Hex	Dec	Name	Hex	Dec	Address
M3PF	Missile 3 to Playfield Collisions	D003	53251				
M3PL	Missile 3 to Player	D00B	53259				
NMIEN	Non-Maskable Interrupt (NMI) Enable	D40E	54286				
NMIRE	NMI reset	D40F	54287				
NMIST	NMI Status	D40F	54287				
POPF	Player 0 to Playfield Collisions	D004	53252				
POPL	Player 0 to Player Collisions	D00C	53260				
PIPF	Player 1 to Playfield Collisions	D005	53253				
PIPL	Player 1 to Player Collisions	D00D	53261				
P2PF	Player 2 to Playfield Collisions	D006	53254				
P2PL	Player 2 to Player Collisions	D00E	53262				
P3PF	Player 3 to Playfield Collisions	D007	53255				
P3PL	Player 3 to Player Collisions	D00F	53263				
PACTL	Port A Control	D302	54018				
PAL	PAL/NTSC Indicator	D014	53268				
PBCTL	Port B Control	D303	54019				
PENH	Light Pen Horizontal Position	D40C	54284	LPENH	234	564	
PENV	Light Pen Vertical Position	D40D	54285	LPENV	235	565	
PMBASE	Player Missile Base Address	D407	54279				
PORTA	Port A	D300	54016	STICK0,1	278,279	632,633	
PORTB	Port B	D301	54017	STICK2,3	27A,27B	634,635	
POT0	Pot 0	D200	53760	PADDL0	270,	624	
POT1	Pot 1	D201	53761	PADDL1	271,	625	
POT2	Pot 2	D202	53762	PADDL2	272	626	
POT3	Pot 3	D203	53763	PADDL3	273	627	
POT4	Pot 4	D204	53764	PADDL4	274	628	
POT5	Pot 5	D205	53765	PADDL5	275	629	
POT6	Pot 6	D206	53766	PADDL6	276	630	
POT7	Pot 7 (right paddle controller)	D207	53767	PADDL7	277	631	
POTGO	Start POT Scan Sequence	D208	53771	WRITTEN DURING VBLANK			
PRIOR	Priority Select	D01B	53275	GPRIOR	26F	623	
RANDOM	Random number Generator	D20A	53770				
SRIN	Serial Port Input	D20E	53774				
SROUT	Serial Port output	D20D	53773				
SIZEP0	Size of Player 0	D008	53256				
SIZEP1	Size of Player 1	D009	53257				
SIZEP2	Size of Player 2	D00A	53258				
SIZEP3	Size of Player 3	D00B	53259				
SKCTL	Serial Port Control	D20F	53775	SSKCTL	232	562	
SKREST	Reset Serial Port Status (SKSTAT)	D20A	53770				
SKSTAT	Serial Port Status	D20F	53775				
STIMER	Start Timer	D209	53769				
TRIG0	Joystick Controller Trigger 0	D010	53264	STRIG0	284	644	
TRIG1	Joystick Controller Trigger 1	D011	53265	STRIG1	285	645	
TRIG2	Joystick Controller Trigger 2	D012	53266	STRIG2	286	646	
TRIG3	Joystick Controller Trigger 3	D013	53267	STRIG3	287	647	
VCOUNT	Vertical Line Counter	D40B	54283				
VDELAY	Vertical Delay	D01C	53276				
VSCROL	Vertical Scroll	D405	54277				
WSTNC	Wait for Horizontal Sync	D40A	54282				

Hardware Register				OS Shadow			
Name	Description	Hex	Dec	Name	Hex	Dec	Address
ALLPOT	Read 8 line Pot Port State	D208	53768				
AUDC1	Audio Channel 1 Control	D201	53761				
AUDC2	Audio Channel 2 Control	D203	53763				
AUDC3	Audio Channel 3 Control	D205	53765				
AUDC4	Audio Channel 4 Control	D207	53767				
AUDCTL	Audio Control	D208	53768				
AUDF1	Audio Channel 1 Frequency	D200	53760				
AUDF2	Audio Channel 2 Frequency	D202	53762				
AUDF3	Audio Channel 3 Frequency	D204	53764				
AUDF4	Audio Channel 4 Frequency	D206	53766				
CHACTL	Character Control	D401	54273	CHART	2F3	755	
CHBASE	Character base address	D409	54281	CHBAS	2F4	756	
COLBK	Color-Luminance of Background	D01A	53274	COLOR4	2C8	712	
COLPFO	Color Luminance of Playfield 0	D016	53270	COLOR0	2C4	708	
COLPFI	Color Luminance of Playfield 1	D017	53271	COLOR1	2C5	709	
COLPF2	Color Luminance of Playfield 2	D018	53272	COLOR2	2C6	710	
COLPF3	Color Luminance of Playfield 3	D019	53273	COLOR3	2C7	711	
COLPM0	Color Luminance of Player-Missile 0	D012	53266	PCOLOR0	2C0	704	
COLPM1	Color Luminance of Player-Missile 1	D013	53267	PCOLOR1	2C1	705	
COLPM2	Color Luminance of Player-Missile 2	D014	53268	PCOLOR2	2C2	706	
COLPM3	Color Luminance of Player-Missile 3	D015	53269	PCOLOR3	2C3	707	
CONSOLE	Console Switch Port	D01F	53279				
DLISTH	Display List Pointer (high byte)	D403	54275	SDLSTH	231	561	
DLISTL	Display List Pointer (low byte)	D402	54274	SDLSTL	230	560	
DMACTL	Direct Memory Access (DMA) Control	D400	54272	SDMCTL	22F	559	
GRACTL	Graphic Control	D01D	53277				
GRAFP0	Graphics for all Missiles	D011	53265				
GRAFP1	Graphics for Player 0	D00D	53261				
GRAFP2	Graphics for Player 1	D00E	53262				
GRAFP3	Graphics for Player 2	D00F	53263				
HITCLR	Collision Clear	D010	53264				
HPOS0	Horizontal Position of Missile 0	D004	53252				
HPOS1	Horizontal Position of Missile 1	D005	53253				
HPOS2	Horizontal Position of Missile 2	D006	53254				
HPOS3	Horizontal Position of Missile 3	D007	53255				
HPOS0P0	Horizontal Position of Player 0	D000	53248				
HPOS0P1	Horizontal Position of Player 1	D001	53249				
HPOS0P2	Horizontal Position of Player 2	D002	53250				
HPOS0P3	Horizontal Position of Player 3	D003	53251				
HSCROL	Horizontal Scroll	D404	54276				
IRQEN	Interrupt Request (IRQ) Enable	D20E	53774	POHMSK	10	16	
IRST	IRQ Status	D20E	53774				
KBCODE	Keyboard Code	D209	53769	CH	2FC	764	
MOPF	Missile 0 to Playfield Collisions	D000	53248				
MOPL	Missile 0 to Player Collisions	D008	53256				
MIPF	Missile 1 to Playfield Collisions	D001	53249				
MIPL	Missile 1 to Player Collisions	D009	53257				
M2PF	Missile 2 to Playfield Collisions	D002	53250				
M2PL	Missile 2 to Player Collisions	D00A	53258				

ANTIC MODES

ANTIC MODE #	2	3	4	5	6	7	8	9	A	B	C	D	E	F
OS MODE #	0	1	2	3	4	5	6	7	8	9	A	B	C	D
DISPLAY TYPE	CHAR	CHAR	CHAR	CHAR	MAP	MAP	MAP	MAP	MAP	MAP	MAP	MAP	MAP	MAP
SCAN LINES	8	10	8	16	8	8	16	8	4	4	2	1	1	1
COLOR CLOCKS	4	4	4	4	8	8	4	4	2	1	1	1	1	1/2
PIXELS/LINE	40	40	40	40	20	20	40	80	160	160	160	160	160	320
BYTES/LINE	40	40	40	40	20	20	10	10	20	20	20	40	40	40
# OF COLORS	2-5	2-5	5	5	5	5	4	2	4	2	2	4	4	2
BYTES/SCREEN	960	800	960	480	480	240	240	480	960	1920	3840	3840	7680	7680

HIGH ORDER INSTRUCTION BITS:

- D7: display list interrupt
- D6: load memory scan register
- D5: vertical scroll enable
- D4: horizontal scroll enable

SPECIAL INSTRUCTION CODES:

- 00-70 blank 1 through 7 scan lines
- 01 JMP: jump (over 1K boundary)
- 41 JVB: jump and wait for vblank

USEFUL OS EQUATES

LABEL	ADDRESS	DESCRIPTION
-------	---------	-------------

PAGE ZERO

- DOSVEC 0A DOS vector
- DOSINI 0C warm start address
- RTCLK 12 real-time clock, 3 bytes
- ATTRACT 4D attract mode flag
- DRKMSK 4E dark mask for attract
- COLRSH 4F scrambles color for attract
- SAVMSC 58 points to beginning of screen data

PAGE TWO

- VDLSLT 200 display list interrupt vector
- VPRCED 202 proceed line IRQ vector
- VINTER 204 interrupt line IRQ vector
- VBREAK 206 software break (00) IRQ vector
- VKEYBD 208 keyboard IRQ vector
- VBLKI 220 immediate vblank interrupt vector
- VBLKD 224 deferred vblank interrupt vector

ROM VECTORS

- EDITRV E400 E: device handler
- SCREENV E410 S: device handler
- KEYBDV E420 K: device handler
- PRINTV E430 P: device handler
- CASETV E440 C: device handler
- DISKIV E450 D: initialization
- DISKINV E453 disk interface
- CIOV E456 central input-output vector
- SIOV E459 serial input-output vector
- SETVBV E45C routine for setting vectors
- SYSVBV E45F vertical blank routine vector
- XITVBV E462 exit vertical blank routines
- SIOINV E465 serial input-output initialization
- SENDEV E468 send enable routine
- INTINV E46B interrupt handler initialization
- CIOINV E46E CIO initialization
- BLKBDV E471 blackboard mode (memc pad)
- WARMSV E474 warm start entry point
- COLDVSV E477 cold start entry point
- RBLKOV E47A cassette read block vector
- CSOPIV E47D cassette open for input vector

IOCB STRUCTURE

RELATIVE ADDRESS	LABEL	DESCRIPTION	SET BY
0	ICRID	handler ID	CIO or OPEN
1	ICNO	device number	CIO or OPEN
2	ICOMC	I/O command byte	user
3	ICSTA	status (errors)	CIO or return
4	ICBAL	buffer address (low)	user
5	ICBAH	buffer address (high)	user
6	ICPTL	put character pointer	CIO
7	ICPTH	put character pointer	CIO
8	ICBL	buffer length/byte count	user/CIO
9	ICBLH	buffer length/byte count	user/CIO
A	ICAX1	D2=read, D3=write on OPEN	user
B	ICAX2	auxiliary information	user
C-F	ICAX3-ICAX6	auxiliary information	CIO

BIT ASSIGNMENTS FOR SELECTED HARDWARE REGISTERS

REGISTER	D7	D6	D5	D4	D3	D2	D1	D0
AUDCX	distortion							
AUDCTL	17-bit poly changes into 9-bit poly	clock Ch 1 w/ 1.79 MHz not 64 KHz	clock Ch 3 w/ 1.79 MHz not 64 KHz	clock Ch 2 w/ Ch 1 not 64 KHz	clock Ch 4 w/ Ch 3 not 64 KHz	hi-pass filter in Channel 1	hi-pass filter in Channel 3	use 15 KHz not 64 KHz
CHACTL						vertical reflect	video invert	video blank
CONSOLE					speaker	option	select	start
DMACTL			display list instruction DMA enable	1=1-line PM 0=2-line PM	enable player DMA	enable missile DMA	playfield	wlctch
GRACTL						latch TRIG0-3	enable players	enable missiles
IRQEN (IROST)	BREAK key	other key	serial input data ready	serial output data needed	SOTF	timer 4	timer 2	timer 1
NMIEN	display list interrupt	vertical blank interrupt						
NMIST	"	"	SYSTEM RESET					
PRIOR	GTIA mode selection		multiple color players	flth player enable	PFO-PF1 P0-P3 PF2-PF3	PFO-PF3 P0-P3 ---	P0-P1 PFO-PF3 P2-P3	P0-P3 PFO-PF3 ---
PORTA,B	right	left	back	forward	right	left	back	forward

### Most Wanted List

#### Dal-ACE Experts

Donny Arnold...216-0766...Call before 10 p.m. 8-bit General knowledge.

Eb Foerster...357-7602...Call from 7 to 10 p.m. Turbo Basic, SynFile, SynCalc, Assembly.

Ron King...(817)283-0674...Call from 5 to 10 p.m. 8-bit hardware.

John Saunders...(817)566-0318 C and Assembler languages.

Michael Trombley...429-6134 ST general knowledge.

Ralph Tenny...235-4035 Call from 7 to 10 p.m. ST general knowledge and hardware.

Rene Tucker 223-6176 8-bit general knowledge.

John Winer...907-1349 Systems Programming and general knowledge.

#### Infomart Directions

From North Dallas, take either Stemmons (I-35E) or the Dallas North Tollway south. From Stemmons, take the Oak Lawn exit, turn East, and park at the Infomart. If you are using the tollway, exit right on Wycliff, go left on Harry Hines Blvd. to Oak Lawn, and turn right. From the South, take Stemmons north, then follow above directions. Infomart is the big, white, steel and glass building south of the other marts. **GUESTS ARE WELCOME!!!**

### Newsletter

#### Advertisements

Personal ads are free to all current members. Please see the editor for details.

#### Commercial Rates

Full Page \$35  
Half Page \$25  
Quarter Page \$15  
Business card \$10

For an additional \$10 per full page, or \$5 per partial page, you can request that your ad be placed on the inside front or back cover or the center page spread. This service is first come, first served.

Ads must be camera ready. Submission deadline for ad copy is the first of the month prior to publication date. That is, November 1st will be the deadline for your ad to appear in the December newsletter. Mail copy to the address on the back page, or contact the advertising manager, editor, or the Vice President of Communications. Copy received after the deadline will be run the following month. For contract advertisers, if no new ad is received prior to the deadline, the most recent ad will be run.

#### Editorial Policy

The editorial staff of the Dal-ACE newsletter reserves the right to edit your submissions for spelling, punctuation, grammar, clarity, and for reasons of space limitations.

### Newsletter

#### Submissions

Submissions are welcome in any form. Submissions can be uploaded to the club BBS or they may be submitted to the editor at the club meetings or by mail.

#### Disclaimer

The material printed in this newsletter reflects the opinions of the authors. Opposing opinions are solicited. Unless otherwise stated, the material in this newsletter is not copyrighted and no rights are reserved.

The purpose of this newsletter is to present information for your consideration. Neither the editor nor Dal-ACE make any claims for the validity or usefulness of this material. The reader is the final judge of any product or advice presented.

#### Infomart Meeting Dates

**Firm Dates:** August 12, September 23.

**Tentative Dates:** October 14, November 11, December 16.

#### Meeting Schedule

10:00-11:00...8-bit SIG  
11:00-11:30...Disk sales  
11:30-12:00...Main meeting  
12:00-12:30...New Users SIG  
....Newsletter Exchange SIG  
12:30-2:00....ST SIG

## DAL-ACE OFFICERS

President:  
Donny Arnold 216-0766

Vice President:  
Brenda Arnold 216-0766

VP Communications:  
Anita Uhl 492-8682

Secretary:  
Michael Duke 739-3116

Treasurer:  
Rene Tucker 223-6176

Members at Large:  
Terry Borchardt 296-4699  
Dave Gramm (214)370-7143  
Randy Randolph 381-7624  
Marc Salas 717-4615  
Jim Jackson 864-0805

## DAL-ACE BULLETIN BOARD

(214) 255-8256

## DAL-ACE VOLUNTEERS

Editor:  
Open

Ad Manager:  
Marc Salas 717-4615

BBS Sysop:  
Jim Jackson 864-0805

8-Bit Librarian:  
Tim Mixon 356-4725

ST Librarian:  
Gary Loges (817)481-4186

ALL MATERIAL PRINTED IN THIS NEWSLETTER MAY BE REPRINTED IN ANY FORM PROVIDED THAT DAL-ACE AND THE AUTHOR, IF APPLICABLE, ARE GIVEN THE PROPER CREDIT. LIKEWISE, PORTIONS OF THIS NEWSLETTER MAY BE REPRINTED FROM OTHER NEWSLETTERS AND ARE SO NOTED.

## DAL-ACE INC.

### Dallas Atari Computer Enthusiasts

Dal-ACE is an independent user education group that is not affiliated with the Atari Corporation. This is the official newsletter of Dal-ACE and is intended for the education of its membership as well as for the dissemination of information about Atari Computer Products.

Dal-ACE membership is \$20 per year. BBS only membership is \$10 per year. This newsletter is written, edited and published by volunteers. Its availability and/or distribution may, at times, be subject to circumstances beyond the control of the club officers. A pink address label indicates that your membership expires this month. Other user groups may obtain a copy of this newsletter on an exchange basis.



P.O. BOX 851872  
RICHARDSON, TEXAS 75085-1872

U.S. POSTAGE PAID  
BULK RATE  
PERMIT NO. 1203  
RICHARDSON, TEXAS 75080

Address correction requested. Return postage guaranteed.

