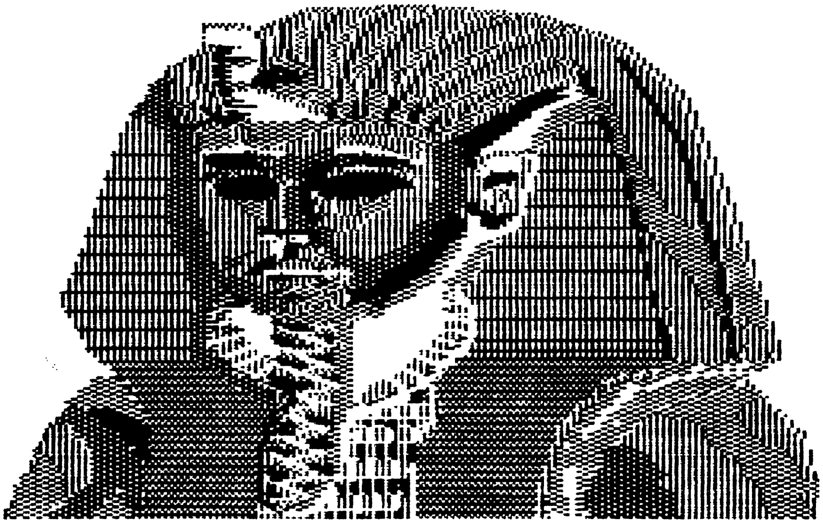
FUJI FACTS

Newsletter of the
Atari Computer Enthusiasts of Columbus

Volume 5, № 2

February, 1987



THIS ISSUE IS FILLED WITH TREASURES!

TUT.PIC
from B/GRAPH

INSIDE THIS ISSUE:

This newsletter is written and published monthly by the Atari Computer Enthusiasts of Columbus (**ACEC**). ACEC is an independent, non-profit organization interested in exchanging information about any and all Atari Home Computer Systems.

The main meetings are held on the second Monday of each month at 7:15 p.m., at DeSales High School (on Karl Road, just south of Morse Rd.), and are open to the public. Other Special Interest meetings are held as announced at the main meeting.

Dues are $12.00 per year, and entitle members to all club benefits (Newsletter, Disk of the Month, Publications Library, SIG meetings, group discounts at selected area merchants, etc.).

Fuji Facts welcomes contributions of articles, reviews, editorials and any other material relating to the Atari computers, or compatible hardware devices and software packages.

ACEC is not directly affiliated with the Atari Corporation; "ATARI" and the "Fuji" symbol are registered trademarks of the Atari Corporation. All other trademarks, copyrights and service marks are registered with their respective owners.

The statements expressed in this newsletter are solely the opinions of the authors, and do not necessarily reflect those of ACEC, its officers or its members. Material contained in this newsletter may be reprinted provided credit is given to both Fuji Facts and the author(s).

The cover of this month´s newsletter was printed with a Star SG-10 dot matrix printer, using XLent Software´s TypeSetter 130. The newsletter itself was printed with a Star PowerType daisywheel printer in elite pitch, using Batteries Included´s PaperClip version 2.0 on a 256K RAM modified Atari 800 XL.

Our permanent mailing address is:
P.O. Box 849
Worthington, OH 43085

PRESIDENT:
Charles Lusco
4624 Channing Terrace, #C
Columbus, OH 43232
863-4016

VICE-PRESIDENT:
Dave Beck
1194 Country Club Road
Columbus, OH 43227
863-8600

PUBLICATIONS LIBRARIAN:
Bill Morgens
4943 Botsford Drive
Columbus, OH 43232
861-2431

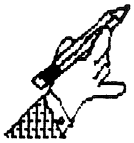
MEMBERSHIP CHAIRMAN:
Mike Compton
1342 Gumwood Drive
Columbus, OH 43229
885-3757

NEWSLETTER EDITOR:
Warren Lieuallen
1652 Hess Boulevard
Columbus, OH 43212
488-3977

DISK LIBRARIAN:
Paul Pelfrey
2199-D Cortside Drive
Columbus, OH 43232
866-1454

TREASURER:
Dave Feeney
2665 Blue Rock Boulevard
Grove City, OH 43123
871-0524

SECRETARY:
Don Bowlin
230 Orchard Lane
Columbus, OH 43214
262-6945



The Editor's Column

Shake, Rattle and Roll!

Although you wouldn´t know it from looking at us (this too will shortly change, we hope), there have been quite a few major changes in the Atari Computer Enthusiasts of Columbus. The January officers´ meeting was the most controversial, the most difficult, and also the most productive we´ve ever had in our four year history. I´m sure the details of all the changes will be announced to you at the proper time, but I´d just like to outline the most visible and dramatic changes.

First of all, to the relief of many I imagine, we have decided to formally split the XL/XE (and 400/800/600/1200, etc.!) group from the ST group. Although still joined under the ACEC umbrella, these groups will now function as separate entities, each with their own officers, their own schedules, and most importantly, their own control over their individual activities. It had gotten very inefficient for a group of "eight-bitters" to discuss and rule on the ST decisions, and vice versa. So, let´s welcome the new Atari ST Users´ Group into existence, and wish them well. For we in ACEC, about all this really means is that the ST Supplements will no longer appear in Fuji Facts, but will instead become the newsletter of the new COSTUG (Central Ohio ST Users´ Group, as I believe they call themselves).

Secondly, and probably to even more people´s relief, our BBS has been moved! After months of mulling over our options, we have finally had the intelligence to select Frank Seipel as our new Sysop. Many of you know Frank already, and know of his own BBS, Pandora. You´ll now get to know both better, as Frank assumes official control of the ACEC BBS, and merges it with Pandora in the future (including a shared message base, and a shared 10 meg hard drive!). I expect our BBS to expand significantly, now that it´s getting some attention again. Please make a note that, because of the move, the phone number will be have been changed as of February first to: **471-8559**.

A new emphasis is going to be placed on the structure and content of our meetings. If you have any ideas or suggestions, please speak to Dave Beck, Charles Lusco or myself. This is your club; let us know how you would like it to serve you!

In this same vein, please consider writing an article for Fuji Facts, or giving a short demo at the meeting. Everyone has something that they can discuss for a few minutes, and you´d be surprised at how useful even the simplest demo can be (no one knows everything). Got a new program or piece of hardware? Got an old, but still reliable? Let us hear about it! I have included an article in this month´s issue to prompt more of you to write things for me. These past two months, I´ve had to rely on CompuServe and other clubs´ newsletters much less than usual, and to that I say, "Keep it up!" Our own original material is much more interesting than a reprint that you could just as easily read elsewhere. I thank you.

I hope you are enjoying the new format of the newsletter. Because of the combination of the offset printing and the true letter quality printer (both of which greatly improve the printing quality), we are able to use this reduced "book" format, and greatly reduce our printing costs. This, combined with our new bulk mail rate (which should be in effect by next month) means that ALL newsletters will now be mailed to you, approximately one week before the meetings (so they should serve as good reminders that the meeting is coming up!). This means that the deadline for newsletter submissions must be moved up to the third Saturday of the month prior to publication, but that shouldn´t be a big deal (For example, if you want something in the March newsletter, I must have it by February 21st. Later than that, and it will go into the April newsletter instead. See? No big deal!).

By the way, if you like what you are reading right now, you might like to know that it´s being printed on a Star PowerType daisywheel printer. Many of you have probably seen this printer at Computer Express. Well, I saw it too, and thought it would really improve the look and readability of Fuji Facts, so.... Watch for a review in the near future.

Until next month, keep those cards and letters coming!

Warren Lieuallen

Antic
On-Line

ANTIC Publishing, Inc.
copyright 1987. Reprinted by
permission from Antic Online.

New XL/XE Products
by Gregg Pearlman
Antic Junior Editor

SUPER RAT

RAOS Zobian Controls
1156 Old Mill Lane
P.O. Box 6406
Wyomissing, PA 19610
(215) 374-5478
48K disk

SuperRAT, a new high-accuracy digital version of Zobian´s mouse for the Atari 8-bit computers, can access every pixel in Graphics 8. It comes with Accu-Draw for $69.95 and with RAOS for $99.95. RAOS ($49.95), the Rat-Actuated Operating System, gives you a "GEM" system on your 8-bit Atari. RAOS has its own icon-based desktop program, Z-DOS, replacing DUP.SYS, and a mouse-controllable, user-programmable windowing environment occupying just 3K of RAM. RAOS uses a Player/Missile cursor to select icons and is compatible with most memory upgrades.

STAR FLEET 1

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171
$49.95, $48K disk

In Star Fleet 1, each new level at the Officers Academy lets you role-play different characters and experience unique tasks. As a member of the Alliance, you must protect its outer regions against the invading Krellan and Zaldron empires. After graduating from the academy, you´ll command one of 36 cruisers with a sophisticated array of weaponry at your fingertips.

BASIC ENHANCEMENTS II

First Byte
P.O Box 32
Rice´s Landing, PA 15357
(412) 627-3596
$24.95, 48K disk

First Byte´s Enhancements to BASIC, Version II is a completely rewritten package, not just an upgrade. You can access many DOS functions in immediate mode, rather than exiting to DOS itself. Enhanced BASIC II makes it easier to trace your program flow for debugging. The software also renumbers, block deletes and provides automatic line numbering.

MIND TUNER

Mind Link Communications Inc.
Box 488 36
Adelaide Street East
Toronto, M5C 2J6 Canada
(416) 961-8858
$24.95, 48K disk

In the market for self-improvement? Mind Tuner uses proven psychological principles to help you manage stress and improve your personality and performance with visualizations, positive affirmations and subliminal messages that you can write for yourself.

VIDEO TITLE SHOP, MERCENARY: 2ND CITY, and FIVE FROM PSS

Datasoft
19808 Nordhoff Place
Chatsworth, CA 91311
(818) 886-5922
48K disk

Develop title screens on your Atari for your VCR or video camera, and then record the screens directly to videotape with Video Title Shop ($29.95). Choose from several type styles and sizes, create borders and add flair with the scrolling and fade-in/out features. The package includes Micro-Painter Plus for graphics that can be used as backgrounds to titles.

Experienced players of Mercenary will find the Second City on the other side of the Planet Targ, and once again you must escape. You´ll be hindered by those warring planetary factions. Mercenary: The Second City ($14.95) must be played in conjunction with Datasoft´s original Mercenary game disk.

Five new games come by way of England´s PSS: Tobruk ($34.95) is a re-creation of the desert battle that threatened England´s presence in North Africa. Bismarck ($34.95) puts you in command of the Royal Navy or the Bismarck itself. Swords & Sorcery ($34.95) is a role-playing fantasy that takes you through the underworld on a quest for secular and spiritual power. Saracen ($29.95) sends you through 100 mazes of increasing complexity as you seek and try to destroy the evil Saracen warrior. Black Magic ($24.95) puts you into 100 mazes of a fantasy world as you set out to destroy the forces that have overrun the kingdom of Marigold, restoring the Golden Age it once knew.

WARSHIP, ROADWAR 2000, GEMSTONE HEALER, WARGAME CONSTRUCTION SET, and SSI CLASSICS

Strategic Simulations Inc.
1046 N. Rengstorff Avenue
Mountain View, CA 94043
(416) 964-1353
48K disk

SSI´s newest release includes Warship ($59.95), a tactical ship-to-ship World War II game, Roadwar 2000 ($39.95), a science fiction adventure, Gemstone Healer ($29.95), the sequel to the role-playing adventure Gemstone Warrior, and the Wargame Construction Set ($29.95), which lets you build your own war, fantasy or science fiction games. The SSI Classics line brings back proven hits -- now at only $14.95 each. Titles include Combat Leader, Computer Baseball, Computer Bismarck, Cytron Masters, Eagles, Fortress, Galactic Gladiators, Gemstone Warrior and Warp Factor.

SOMEONE REALLY SHOULD WRITE SOME REVIEWS OF ALL THESE NEW 8-BIT PRODUCTS, DON'T YOU THINK?

HOW ABOUT YOU?!

New Comers' News

Reprinted from Current Notes, newsletter for the Washington Area Atari Computer Enthusiasts, Dec 86.

WORD PROCESSING

By Ron Peters

Last month we discussed DOS and how it helps you communicate with your disk drive. Now let´s talk about word processing and what it does.

Word processing is a fancy name for a "process" that includes a typewriter, a printer with different fonts (styles of type), scissors, paste, a wastebasket, and a Xerox machine. Sound complicated? It really isn´t; let´s have a look.

First of all let´s define "fonts." A FONT is just another name for a typeface, or a style of print. These include Roman, Helvetica, Press, etc. (A good example would be found in the Print Shop program).

These various fonts can be found in familiar sizes of pica, elite, etc., which refer to the number of characters per inch; pica contains ten c.p.i. while elite contains 12 c.p.i. Your printer manual will discuss these in detail since some printers are capable of additional sizes. Thus a Roman typeface is a name given to a style of print, where all the letters are designed in a certain way.

Depending on the brand of your printer, it may be possible to print in pica, elite, condensed, proportional, expanded, superscript and subscript typesizes. The style of print will vary between brands, or your printer may be able to print different fonts that can be "loaded" into it. Your printer manual will give further details.

Let´s get back to word processing. This is where your typing appears on your computer monitor screen before being put on paper so that you can easily change the text to make corrections electronically. We´ll compare this to a typewriter.

Pretend that we are writing a letter to the IRS explaining why we deducted $1,200 for vitamins for our pet pooch, Ralph. Let´s assume that you type this letter on your typewriter, and, when it´s finished, you wished you made the margins wider. No problem! With your word processor you just issue commands to change the margins and reprint the letter with a very minimum of effort. The program automatically reformats the text to accomodate the new settings you just entered. Try this with your typewriter!

Now, on that same letter, you forgot to include a sentence in the middle of the letter explaining that Ralph really is a member of the family and it´s not fair that he is not covered by your family health insurance (that´s why you took the full $1,200 deduction). Again, no problem. Just use the "insert" command and place the sentence anywhere in the document you´d like. The word processing program will make the necessary adjustments to the text to allow the sentence to fit. Of course, if you later decide that you´ll be pushing your luck with the IRS, you can easily delete the sentence, and the program will automatically fill in the "gaps" created by the deletion.

Looking at your finished letter, you now discover that the second paragraph (where you itemize the $1,200 for the Count Chocu-pooch vitamins) should have come after the paragraph where you make a case for Ralph being a bona-fide dependent (and a full-time student, even though he´s 96 years old by human standards). With a typewriter you would either have to re-type the whole letter or cut up the one you´ve just finished, paste the paragraphs in the correct order, and then Xerox the whole mess.

However, with your trusty word processing program, you easily do a "block move" and put the paragraphs in the right order. And, all in less time than it takes to read this.

If your word processing program is really "state-of-the-art," it will include a spell checker that will automatically read your masterpiece and check for errers (like that one), and even give you a dictionary search or synonym option.

What else will it do? If what I´ve just described is not enough to make you throw your typewriter in the wastebasket (no, that´s not the wastebasket feature - that´s coming up), then read on. A good word processing program will include such features as:

* double column printing (such as you are now reading)

* ability to mix pica, elite, condensed, etc., typesizes

* automatic page numbering

* page "headers" and "footers" (automatically puts the same text at the top or bottom of each page)

* superscripts and subscripts (for bibliography references, etc.)

* file merge (ability to automatically put individual names, addresses, etc., on form letters)

* ability to underline

* ability to doublestrike (print over each letter for emphasis)

* change margins at the top and bottom of the page

* change the form length (size of paper you are using)

* format the text in "block right" (even margins on both sides, like a newspaper column) by automatically adjusting the text

* plus many other features that space doesn´t permit mentioning.

Of course, you can save these masterpieces on a floppy disk (most word processing programs have a form of DOS built in) and later retrieve them for editing (making changes or corrections) or printing.

Or, (and here comes the wastebasket feature) you can delete these documents, (like when you get an "ARE YOU JOKING?? response from the IRS) to make more room on your disk, or just to get rid of the evidence.

Before you buy a word processing program, check out the features compared to other programs, but don´t make the mistake of buying more features than you can ever use. (You might try borrowing one or several different brands of word processing programs from a friend to help in your evaluation.) For example, a program that produces an automatic table of contents isn´t worth the extra cost if you are only going to write anonymous crank letters to the IRS.

Myself, I use AtariWriter + and it does everything I want and more; all for around $39.00. (Myself, I use PaperClip 2.0, and it does everything AtariWriter + does and more; all for around $39.00! - Ed.)

Next time we´ll talk about "spreadsheet" programs and what they can do for you.

Book Review

INSIDE ATARI BASIC
by Charles Brown

Here I go with another one of my book reviews. I sure hope that someone is getting some good out of them! In the SIG meetings, I have tried to show people what BASIC programming is all about. As with any group of more than two or three, I have found that there are some people at these meetings that have some very simple questions. These are just as important as the technical ones, and their answers are often more informative.

Also, in going to different stores, I have noticed people asking about books that teach Atari BASIC. I believe they the people asking these questions are just starting out, and need something to teach them the basics. So, I am going to tell you about one particular book that I feel will help you out. It is very good for the person just starting. You might be startled to know that I find myself going back to it, too.

The book is called INSIDE ATARI BASIC. It is published by Reston Publishing. I feel that it gives some very good help in starting out on the computer and how to program it. It even starts out with how to put your cartridge in and turn the computer on. It gives some examples of possible problems and how to correct them.

Then, the book gives a pretty good explanation of the keyboard. It starts out with the simple keys. They are the system reset, delete, and return keys. While they may be simple, they sure are important.

It then shows how to get the quotes on your screen (shift-2), how to use the Caps and the Shift-Caps keys for lower and upper case characters, and how to use the inverse video key in conjunction with Caps. Then the book shows how to make your computer do the simple math functions. It then goes on to one of the most important functions, which is printing something on the screen.

The book also goes on to show the difference between the direct and programming mode. Some people call the programming mode the deferred mode, and the direct mode, immediate.

In the direct mode, you type in a command and hit the return key. The command will be executed instantly. In the program mode, you first put in a line number followed by the command(s). The command(s) won´t be executed until you RUN the program. In the discussion in the book it goes on to tell about the line numbers and program lines. It even shows how to replace a line or erase it completely.

Then the book goes through most of the commands that you will be using in Atari BASIC. It gives a pretty simple explanation of each one and examples on how they are used. Then it shows how to use the control key. It even has some examples with step by step instructions on what to do. Then it goes into what variables are and how to use them. It gives a brief introduction to string variables and what they are used for. It even shows you about an arrays and a matrix.

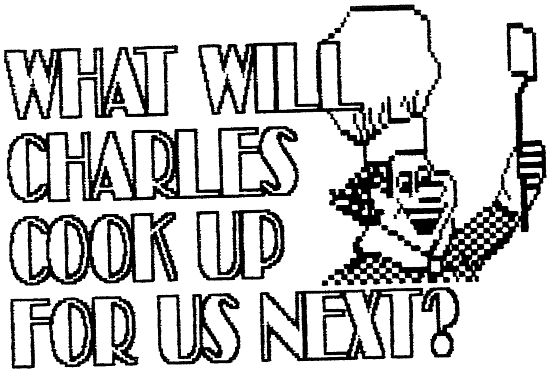
The best feature I feel this book has is it´s explanation on the Atari graphic capabilities. It shows how to use the control keys to create the graphic characters on the screen. One of the nice features that I use is to see the different graphic modes and their sizes. I am always forgetting how many rows and columns are in each mode. I like to avoid the famous error 141. The book also shows a sample of each mode and what it looks like. I feel that it could be copied and used to format your displays, especially the graphic modes 1 and 2 used for putting the larger text on your screen.

Along with showing the different modes, this book gives a real good explanation of using color in your programs. It explains the COLOR and SETCOLOR commands and how to use them. It even shows the default colors for each register in each mode. I have not found an easier source for that information. It tells which color register is used for each function in each of the different modes. It even shows what the default color for each register. The most important thing for me in this book is the color chart it has. It shows the relationship between using the values in the COLOR or SETCOLOR commands and the values you can use to POKE the color directly to the screen color locations. For example, if you wanted to have the darkest red for your screen color in graphic mode 0, you would use SETCOLOR 2,4,0. Or you could use POKE 710,64. Either command would have the same effect. In other words, the number 64 would be the color 4 with a luminance of 0 -- these are the sorts of numbers you can get out of this chart of color values. I find this chart very useful. This book even shows how to use the GTIA modes.

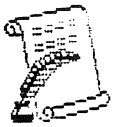
This book also gives a little information on using the sound commands, along with a chart of the numbers which correspond to the notes in the scales our Atari is capable of.

Inside Atari BASIC also delves into the use of the PEEK and POKE commands. In doing this, the book also tells about some of the memory locations and what they are for, since there are a number of things the Atari can do, but only by using a POKE. Some of these locations are the color registers, the margins, and the cursor control. It even shows how to run the program recorder from your computer.

This book dose have a lot of helpful information. I think that it is badly organized, in that it wants to jump around from one subject to another. For example it goes from graphics to color, then to sound, then to PEEKs and POKEs and memory locations, then back to graphics and color. I think they could have planned the book layout better. Nonetheless, if anyone is looking for a good book to help them understand BASIC, they should consider this one. It's one of the better ones, and has more than its share of staying power. I still go back to it every once in a while



READ THE NEXT
ISSUE OF
FUJI FACTS
TO FIND OUT!



Advice to Authors

The ACEC School of Journalism
reprinted from the JACG Journal
by Donald Forbes -- JACG

Welcome to the ACEC School of Journalism. Upon completion of this six-month course, you can consider yourself a full-fledged writer! We are offering this course as a service to our readers (viz., to give them something to read!).

Your assigned textbook will be your back issues of the ACEC Newsletter (Fuji Facts), a national leader in its field. Your instructors will include such familiar journalistic names as: Chris Crawford, Keith Ledbetter, Charles Brown, Warren Lieuallen, Bill Morgens, and many others.

Prerequisites for the course are: a word processor (often referred to as a yellow pad and a #2 pencil outside of computing circles!), and a desire to see your name in print.

To graduate, you will be required to submit six articles (one a month) to the Fuji Facts editor (That´s me! - Ed.). A variety of different forms will be acceptable, in this order of preference: PaperClip file on a disk; PaperClip file via modem or BBS; any other Atari word processor file on disk, modem or BBS; hard copy; verbal!.

ASSIGNMENT ONE

Your first assignment will be to write an article on "My Favorite Atari Game". Dream up a catchier title. Then,. mentally construct your lead paragraph (Note leading paragraphs in your favorite paper. They attempt to answer the famous W´s: What, Where, Who, When and Why, and often hoW much?). Try to have your first paragraph really grab the reader´s attention. Arouse his curiosity

Once this is clear in your mind, think about it for a day or so, and then sit down to write. If you´d like a quick and dirty outline, just use a spreadsheet.

Write your article all at once, without interruptions (although you can insert reminders to yourself to check or add certain facts or spelling [Spelling is not a big deal, as I´m now using PaperClip 2.0, with SpellPack in RAM! - Ed.]). This procedure saves hours of wasted time staring at the keyboard, and makes your story look fresh.

The next day or two, re-read your story to check and double check the facts, and to fix any clumsy, awkward phrasing. On no account attempt to rewrite or significantly revise your original text. Don´t fall into the endless revision trap. If you really think you could do better, start over again from scratch.

Every actor needs a good exit line before he leaves the stage, and so does your article -- if you can find on. With or without this closing gem, print out your article, and bring it to the next meeting. Get some feedback on it from some of the officers, and show it to the editor to make his day!

ASSIGNMENT TWO

Write a review of your favorite non-game program, or another game if you´d rather.

When constructing your article, note the structure of the stories in the paper. After the main facts are stated in the first paragraph, the remaining paragraphs develop the points made in the the lead section. The structure of the paragraphs looks like sets of inverted triangles, like this:

```
----
\  /
 \/

----
\  /
 \/
```

The reason for this: newspapers always have to be prepared for a new story to break. An old story can always be shortened by cutting away from the bottom. The top part of the story will always make sense, nomatter how many paragraphs are chopped from the bottom.

In a program review, you should devote most of the review to an explaination of the program´s operation. Let the facts speak for themselves. Then, wrap up your views in a final paragraph. Be constructive, but critical if it is deserved. If the program is really worthless, it probably doesn´t deserve a review (and instead should simply be given a warning so others don´t buy the same junk).

ASSIGNMENT THREE

Describe a program which solves a problem. Pick a problem which can be solved by a BASIC program that you have (or could have, if you really wanted to!) written yourself, and then show how the code provides an answer. Other good sources are the many books and magazines which feature type-in programs.

The objective here is clear writing: spell out the problem and the solution in clear language so that a non-expert can follow the argument. Every how-to book is written in the clear, precise style of a kitchen cookbook. Always write in a logical pattern. This saves the reader the trouble and annoyance of unraveling tangled prose, which only hides the solution to the problem. Readers looking for poetry or philosophy will not seek out Fuji Facts.

ASSIGNMENT FOUR

Write a book review. From your bookshelf, a friend, or the ACEC publications library, find an Atari book and then try to present in perhaps one page or so the message the author has for his readers (including you!). You should be fair both to the author and to the reader. Let the author present his views, and at the same time, let the reader judge whether the book has worthwhile information based on your own views.

It´s usually best to reserve your personal views for the final paragraph. Alternatively, you can straddle the fence, and present your opinions immediately after each of those of the author.

ASSIGNMENT FIVE

Write a condensation of a computer article. Find an article that would interest our members. Then, condense it to a page or less. The challenge here is to have your article stand on its own, so that the condensation process does not show. Your first paragraph, of course, should cite the original article, and the fact that you condensed it.

As a writer, you should be aware of the copyright law, which was designed to reward a writer for his labors, but was never intended to prevent the spread of knowledge (Read it in the library, especially if you write for money). The law makes it clear that one cannot copyright an idea, only the presentation and expression of the idea.

Copying less than 300 words is not considered a copyright violation. In addition, the law provides exceptions for educational institutions and for "fair use". Every dictionary copies definitions from other dictionaries (there are only so many ways to define the meaning of a word). Furthermore, in a suit, the original author just show that he has actually suffered damages.

Your main concern, of course, is to protect your own reputation and to avoid embarrassment to your editors (That´s me again! - Ed.).

ASSIGNMENT SIX

Write an article on your own field of expertise. Here again, you must visualize your reader. You have to capture his attention from the beginning, appealing to his curiosity, fear, greed, hunger or his desire for news, information, understanding, attention, self-improvement or whatever! Once he starts reading, you must relentlessly hold his attention with tight prose (no digressions) and well constructed short paragraphs. The first sentence in each paragraph should let the reader anticipate what will follow in the rest of the paragraph. If the article is well constructed, the reader should be able to pick out the whole story by just reading the first sentence of each paragraph. Then, go outdoors and read the article aloud as if you were giving a speech. If it doesn´t sound like you, fix it.

GRADUATION

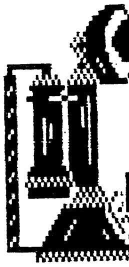
It should be clear by now that writing, like any other task you´ve ever had, has three parts that should never be overlapped: preparation, execution and clean-up.

Preparation: First, organize your story in your mind. Your non-waking mind will help you, if you just learn to listen. Do not write a single word until you have resolved all your uncertainties about the article.

Execution: Write out the whole story all at once. Aim to be a fast writer, not a slow one. Never edit while you are writing.

Clean-up: After a day or so to cool off, re-read and tidy up all the loose ends. Check all your grammar and spelling, and make sure that in the end, you are conveying the information you really wanted to present.

After completion of this program, you will not only be a much better and more efficient writer, but you will also have made Fuji Facts a much more interesting and valuable newsletter. What have you got to lose?



In-Depth Tutorial

ANTIC Publishing Inc.
copyright 1985. Reprinted by
permission from Antic OnLine.

CHRIS CRAWFORD′s ASSEMBLY LANGUAGE COURSE - Lesson 5 of 8

LESSON FIVE: INDEX REGISTERS & LOOPING

We are now going to expand the model of the 6502 that you have been using. Until now, the 6502 I have described had nothing more than a status register, program counter, and accumulator. Now I am going to reveal the existence of two new registers in the 6502: the X- and Y-registers.

These two registers are eight-bit registers just like the accumulator. You can load numbers into them and store them out just as you can with the accumulator. You cannot do arithmetic or Boolean operations with them as you can with the accumulator. But you can do a number of very special things that greatly increase the power of the 6502.

Let′s start with the simple move instructions. The first are LDX and LDY, which load the X- and Y-registers the same way that LDA loads the accumulator. Then there are STX and STY, which store the X- and Y-registers the same way that STA stores the accumulator. There are also four commands for transferring bytes between registers; these are TAX (transfer A to X), TAY (transfer A to Y), TXA (transfer X to A), and TYA (transfer Y to A).

Then there are four special instructions that you will use very often. These are INX and INY, which increment (add one to) the X- and Y-registers, and DEX and DEY, which decrement (subtract one from) the X- and Y-registers.

Finally, we have the CPX and CPY commands, which compare X or Y with the operand of the instruction. These two instructions operate in exactly the same way that the CMP instruction works, except that they use the X- and Y-registers instead of the accumulator.

What are these two registers used for? Well, they are sometimes used as temporary registers. If you are in the middle of a lengthy computation, and you need to save a value currently in the accumulator to make room for something else, the X- and Y-registers are a handy place to stuff values away for temporary storage. Programmers do this all the time.

However, temporary storage is not the only use for these registers. Their real purpose and value arise from their utility as index registers. Index registers go hand in hand with loops; the best way to show you how they are used is to dump the whole schmeer at once and then explain it.

So consider the following problem: your program has to deal with the possibility of user errors. Suppose you require the user to type in a file name for your program to read. What happens if this file is not on the disk? You have to put an error message on the screen that says, "FILE NOT′ ON DISK!" How do you print the message? Here′s a sample bit of code that will do it:

```
       LDX #(ENDMSG-ERRMSG-1)
LOOP1  LDA ERRMSG,X
       STA SCREEN,X
       SEC
       SBC #$20
       DEX
       BPL LOOP1
       JMP ELSWHR
ERRMSG DB ′FILE NOT ON DISK!′ ENDMSG
```

Let´s take apart this code and explain it step by step. First thing we do is load the X-register with the number of characters (minus one) in the message. The expression (ENDMSG-ERRMSG-1) will calculate that length at assembly time. This turns out to be 17 characters long. If we were pedestrian about it we could have just written LDX #16, but this way, if we decide to change the message we don´t have to remember to go back and change the LDX command. Neat, huh?

OK, so now we have a 16 in the X-register. Now the 6502 comes to the next command -- LDA, ERRMSG,X. This command tells it to load the accumulator with the byte at (address ERRMSG, indexed by X). What this means is as follows: the 6502 will take the address ERRMSG and add the value of the x-register to that address. It will then go to the address so calculated and load the accumulator with the contents of that address. Since X contains a 16, the 6502 will go to the 16th byte after the first byte in the table ERRMSG. If you count characters, you will see that the 16th byte is the exclamation point. So the 6502 will load the ASCII code for an exclamation point into the accumulator.

The next two instructions (SEC, SBC #$20) are necessary to correct for the Atari´s nonstandard handling of ASCII codes. They make sure that the exclamation will be printed on the screen as an exclamation point.

The next instruction (STA SCREEN,X) stores the result indexed by X. The 6502 will add the contents of X (still 16) to the address SCREEN. It will then store the contents of the accumulator into that address. If that address is part of screen RAM, then you will see an exclamation point appear on the screen.

The next instruction that the 6502 encounters is the DEX instruction. This instruction subtracts one from the X-register, making it a 15.

Next, the 6502 comes to the instruction BPL LOOP 1. This will branch if the N-flag is clear. The value of the N-flag is affected by a DEX instruction. The value of bit 7 of the result is transferred to the N-flag. Bit 7 of 15 is a zero, hence the N-flag is clear, hence the 6502 will indeed take the branch. Note that it branches back up to LOOP 1.

Now it will repeat the process, only this time X contains a 15, not a 16. It will therefore grab the 15th character, an ASCII ´K´, and store that to the screen position just before the exclamation point. Then it will subtract one from X to get a 14, and will continue the loop.

This process will continue, with the 6502 grabbing bytes in reverse order from the table and storing them onto the screen, until after the 6502 does the zeroth byte. When X contains a zero, and the 6502 executes a DEX, it obtains the result $FF. This sets the N-flag. When the 6502 encounters the BPL command, it will NOT take the branch; instead, it will skip the branch and go on to the JMP statement. The loop is terminated.

In this one fragment of code you have seen two major ideas: indexed addressing and looping. They are so closely related that it is hard to talk about one without talking about the other.

You can use indexed addressing with either the X-register or the Y-register. You most commonly use indexed addressing with the LDA and STA commands, but you can also use it with many of the other 6502 commands: ADC, SBC, CMP, AND, ORA, EOR, LSR, ROR, ADL, and ROL can all be used with indexed addressing. Indexed addressing allows you to work with tables or arrays of data.

There is one ugly catch: all of your arrays must be less than 256 bytes long, because the index registers are only eight bits wide. Most of the time this is not a serious problem. However, if you must address a larger table or array, you can use indirect addressing. To do this, you calculate the address that you desire to access, store that address in two contiguous bytes on page zero (low, then high) -- we call these two bytes a pointer -- and then refer to the pointer like so: LDA (POINTER),Y

This instruction will take the address out of pointer, add the value of Y to it, and load the accumulator with the contents of the address so calculated. If POINTER contains $4567 and Y contains a 2, then the 6502 will load the acumulator with the contents of address $4569. You are still restricted by the size of Y, but you can always go back and change the POINTER if you need to span larger arrays. In this case, you frequently just leave Y equal to zero and do all of your indexing directly with changes to POINTER.

The last topic I will take up is termination techniques. Every loop must somehow be terminated if you are to avoid the problem of the Sorcerer´s Apprentice. You will note that the programming example I gave used a rather odd approach. I started at the end of the array and worked backwards. Why not start at the beginning and work forwards? It´s slightly more efficient going backwards than forwards. When you go forwards, you have to terminate the loop with:

```
    INX
    CPX #17
    BNE LOOP1
```

Whereas when you go backwards, you need only use:

```
    DEX
    BPL LOOP1
```

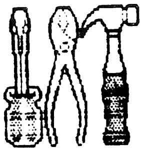
Going backwards you save one instruction. However, if this confuses you, feel free to count forward; that works, too, only it´s a little less efficient.

There are lots of other sneaky ways to terminate loops, but they fall into advanced topics! (Isn´t this advanced enough?! - Ed.)

COMING NEXT MONTH:

SUBROUTINES AND THE STACK

IN THE ASSEMBLY LANGUAGE SERIES BY CHRIS CRAWFORD



Hardware Review

THE ATARI TOUCH TABLET
by Charles Brown

In past articles I have tried to review a number of different programs. I have always tried to concentrate the newer ones, that is, the ones that have just been released. These are ones that you may have seen advertised, and have wondered what they are all about. I realize that there are many older programs out there as well. These may also be some that you may have considered, or, there maybe some new users out there who might wonder what these programs are about. (A review of any program, new or old, is always welcome in Fuji Facts - Ed.)

For a change of pace, in this article I will try and review the Atari Touch Tablet and the AtariArtist cartridge that comes with it. The tablet is plugged into the first controller jack of your computer, just the same as if it was a joystick.

The touch tablet is used to interact with the area on your television screen or monitor. The rectangular area of the tablet is meant to approximate the screen itself -- draw on the tablet, and you are drawing on the screen. The tablet has two buttons, one on each of the two upper corners. It also has a stylus (pen) used for drawing and menu selections, which has a separate button on it (Unlike the Koala Pad. This is a nice feature - Ed.). The buttons are used as triggers to activate the tablet functions or make menu selections. These buttons are just like the buttons on your joysticks or paddles. In fact you can use the tablet to replace either the joystick or paddles in some games (Actually, very few work, because the tablet itself uses the values for paddle 0 and 1, the tablet buttons are the paddle triggers, and the stylus button is joystick 0´s trigger. Very few programs are compatible with this format. If anyone is interested, I can provide a list of how to use this feature within your own BASIC programs - Ed.).

One nice feature is the lock option. To use this, first touch the stylus to the tablet surface. Then push the right button on the tablet. This will then "lock" the button in the on position. This way you can start drawing and lift the stylus up to see what your work looks like. Then you can put the stylus back on the surface to resume your drawing. In this way you can draw without constantly holding down one of the three buttons. It should keep your fingers from getting tired! When you want to turn this feature off you just simply push the right button of your tablet a second time.

The tablet itself is well constructed, and comes with a plastic overlay for the drawing surface. This helps to protect the surface, and is also handy for slipping pictures under for tracing purposes, etc.

The AtariArtist program comes in a cartridge. This makes for instant loading instead of waiting for it to come off the disk drive. If you wish to save your pictures off to disk, however, you should boot the program up with a DOS disk in drive one. This is because you need to have the disk operating system in memory for the computer to recognize the disk drive.

Once you have the program in, pushing the start key should take you to the main menu. This gives you a wide variety of options to choose from. There are 14 drawing options to use. Then there is a storage option used to store your creations. Most of the options are quite easy to understand, especially if you have used some form of drawing program before. Some of the features are Line which is used to automatically draw a line between to points. Then there is Circle and Disk that you can use to draw a perfect circle, either "hollow" or filled in. The Frame and Box feature is quite similar, drawing rectangles instead. These options are quite helpful to reduce the amount of work which must be done freehand. Next, there is the Ray feature. This one lets you create a fan type display of lines.

One of the most important features is the Fill option. It lets you fill an area of your picture with a predetermined color or pattern. This again is so you don´t have to do it freehand. There are two "support" drawing functions. They are the Mirror and Magnify. The mirror option lets you draw in one area of the tablet, and will reflect the drawing to other symmetrical areas of the screen. You can choose from horizontal, vertical, diagonal, or four way mirroring, or you can turn off the feature. Let´s say that you choose the vertical option. You draw a line going up on the top part of the screen. You will also have a line going down on the bottom part of the screen, produced in the "mirror" (I have found this feature takes some getting used to - Ed.).

The second support function is Magnify. This enlarges a portion of your creation eight times it´s original size. This very useful in making corrections, erasing small areas, or detail work.

The storage functions let you access your external devices, which would be either the tape recorder or disk drive. Here you can either save or load your work to the device. It also lets you delete a picture file from disk. You can even format a disk from within the program. It will even give you an index of picture files (a directory). This is a very nice feature. All you have to do is get the index of files listed on the screen. Then, you simply point your stylus to the file name that you want and push one of the control buttons. This way you don´t have to type the file name in from the keyboard.

You are even given a choice of brushes to use. You have 9 different brushes to choose from. They come in a variety of widths and shapes. You have a horizontal, vertical or diagonal line brush. The diagonal brush can produce some pretty good effects, especially when drawing up and down and sideways.

In the color selection, you have your 4 basic solid colors and 12 different patterns to choose from. They can be used for any of the different types of drawing functions or for the fill feature. As with the other options on the menu, you just have to touch the stylus to the color area on the screen, and push a control button to select it. The patterns are a combination of your 4 solid colors (including your background color).

The most fascinating feature is the color menu. Here you can choose from the 128 colors that the Atari has. The nicest part is the color chart it has. Here you just pick what color you want to change directly on screen by using the two charts provided. The chart on the left lets you change the actual color (blue, green, red, etc.). The one on the right lets you change the brightness (dark to light). This way you can tell at a glance all the different colors that are available. If you like, it will send you back to the colors you had before making the changes. You can even go back to the default colors that you had when you turned the program on. Another nice feature is that you can have a rainbow display on your screen for any one of the colors.

One of the more useful features is to change your colors right on your picture. Just choose this option and touch the stylus to one of the color areas on the picture and push a control button. You then move the stylus up or down to change the color level (either dark or light). Moving the cursor left or right will change the actual color. This way you can see instantly how the color changes affect your picture.

In short I feel that this is a very useful program. Although it is limited by your inherent artistic abilities, it has a lot of different features that you can use. I have always wanted to be an artist, but have never had the ability to put my ideas down on paper. With a program like this, I can do my drawings and make any changes that I need easily. If you have never seen the Atari Touch Tablet before and are considering a artistic program, you should consider this one.

Book Review

THE ATARI BASIC SOURCE BOOK
A COMPUTE BOOK PUBLICATION
by Charles Brown

Once again I am going to try my hand at reviewing a book. Although I never have been much of a reader, I do consult a book or two in order to get some facts. For example, I am now working on a machine language program. I need to know the file number for a file which the user selects. I figured I would have to do a search through the directory sectors in order to find the proper name, and then figure out the file number by knowing what sector the program began at, and the offset into it.

Instead I looked in the MAPPING THE ATARI book, also by Compute!. I discovered that the operating system saves the file number off in the file control block. By looking at this location, I would simply get the file number I needed instead of having to search for it. So all these books do serve a purpose.

I am going to review the Atari Basic Source Book by Compute!. I know that because I have done several articles on machine language, some of you may think that I have ignored BASIC. So, I thought that I had better be getting back to it.

This book gives a detailed explanation of how BASIC works. When you read it, you also get a good look on how the computer works, because aside from giving an explanation of BASIC, this book also gives the source code for the Atari BASIC language.

The first part of the book explains how we communicate with our computers. It tells a little bit about machine language programing. This is important because BASIC is an extension of machine language. It tells why we have to have an assembler or translator to go from a source code to the raw machine language instructions. When reading this book, you will find that there are two different ways of translating a source code into object code. The first way is a compiler; the second is an interpreter.

A compiler is used by machine language programming as well as several other languages, such as Action!. First the user types in a source code. Then they run the source code through an assembler or compiler. This translates the source code into a separate object code. One advantage of this method is speed -- once the computer has the object code, it can execute it´s instructions immediately. The problem is that during the assembly process, if there is an error you will be stopped and have to go back to the source code and correct it. Depending on your skill as a programmer, this three-step debugging can be very tedious. Also, just because the source code made it through the assembler does not mean it is going to work. While executing the code you may still have mistakes, which will also need to be fixed and then re-compiled.

Atari BASIC, on the other hand, uses

the interpreter method. Under this method, you as the programmer type in the source code. Then, each line of your program is translated to object code while the program is running. This way if you do have an error, you have the source code right there. You just make the changes and run it again. The only problem here is that each time you run the program, each line of the source code has to be translated. As a result it is much slower in execution (Compare the difference between BASIC and ACTION! - Ed.).

The next section of the book tells you about the design of Atari BASIC. It explains tokens and what they are for. It also tells about the different tables that are used. There are two types: ones that are in ROM (read only memory), and ones that are in RAM (random access memory). It also tells about the different parts of the BASIC language cartridge. They are: the pre-compiler, the program editor, and finally the program execution area.

The next section tells about how BASIC uses memory. In this part, you get more explanations on the different tables used by BASIC. It goes through each table and explains the data in it and how it is used, as well as its relationship to the data in other tables. Without all of these tables, BASIC would not be able to work.

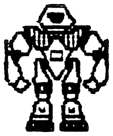
The next sections give a detailed step by step look at the editor, the pre-compiler, and the executor. In these sections you will see exactly what BASIC has to do in order to run your program. It also shows how it uses the different memory locations, much like the Atari Memory Map (also by Compute! - Ed.) does. It tells a lot about the different routines built into our computers.

The next sections take you through each of the commands. Here you will see just what BASIC has to go through to run your program. You can see that the computer has to do a lot of jumping around from place to place! In showing you the different routines, it gives the hexadecimal address of the routines, which may be of value to you.

In the last sections of the book you are given some BASIC programs. These programs will allow you to look at the different goings-ons in BASIC. Some examples are how to look at different memory locations, or listing your variables and tokens. Some may find these programs useful, although these are advanced topics.

As I mentioned earlier, the last section is the source code. In this part you can really see the relationship between BASIC and machine language. I have glanced through it and found it does have some good remarks, although there still are not enough comments to really explain what is happening to anyone other than an experienced machine language programmer. Perhaps I still need to learn more about the computer. I have said before that I am no expert.

I hope that this article will give you a little insight into what this book is about. It does go into great detail about everything that it covers. I feel that it was basically written for someone who has a good deal of understanding about the computer, and would like to get even more. I do feel, though that this book could help everyone at least a little.



Hardware Modification

130XE/800XL Battery Back-Up System
Shareware Hardware Plans by:
Pete Hunter

This battery back-up system is designed to work with the Atari 130XE or 800XL computer. If you are running a BBS on an expanded memory Atari, this will allow you to run the message bases in the Ram-Disk without fear of losing them due to those little power "blips" and those occasional long outages. The little 10 second power losses are the most frequent cause of heartache to a BBS SysOp.

Well, fear no more. This hardware project has been several months in the making and testing. I have been using it on my BBS (a BBS:EXPRESS! system) for about 3 months now. It´s best advantage is that you don´t even have to open the case on your computer, as it requires no alteration to your computer what so ever.

At first I thought about offering it to Antic or Analog for a hardware project, but then decided to release it as a Shareware type hardware plan. As you should already know, Shareware really isn´t Public Domain material. If you use these plans and they work for you, any donation you care to make will be appreciated, and might encourage me to design a battery backup for the MIO board by ICD, which would certainly be cheaper than a UPS transverter type system for about $200.00 or more.

These plans can be built for about $25.00 from Radio Shack parts. The author (and editor! - Ed.) accepts no responsibility for them, due to inability to control user design, techniques and workmanship.

Figure One
Circuit Diagram

```
        |----------|
        | 12 Volt  |
   -<<<<o Battery  |
    |   |----------|
    |              |
    |              |
  Ground    o--o---o+Positive
            |  |   |
            |  |D1
         R5 /  -   Voltage Regulator
            \  |      276-1770
            /  |        7805
            o--o  |___|___|
     1N5400    |  |   V1  |
     276-1101  |  |In     |Out    +5v
    o-->|---o--o--o----o--o---o--o--|
    |       |  |  | Gnd|      |  |
    |       |  |  |    |      |  |
    |  C1   |  |  =C4  |      \  =C5
    o--|(---o  |  |.1uf|    R4/  |
    | .01uf |  |  |    |      \  |
 T1 |       |  |  |    |      |  |
o)|(o       |  |  |    |      |  |
 )|(---Gnd. |  |  o----o--o---o  |
o)|(o       |  |       |  |   |  |
|   |       |  |       |  |   |  |
1   |  C2   |  |+    R1\  \   \R3|
1   o--|(---o  =C3     /  /R2 /  |
7v  |.01uf  |  |4700   \  \   \  |
    |       |  |uf     |  |   |  |
    |       |  |       |  |   |  |
    o-->|---o  o-------o--o---o--o-|
    276-1101   |                  Gnd.
    1N5400    Gnd.
```

Figure Two
Power Plug connection to computer
Pin Configuration

```
 7.    .6
3.      .1
 5.    .4
    .2
```

1. +5 Volts 2. Shield 3. Ground 4. +5 Volts 5. Ground 6. +5 Volts 7. Ground

NOTE: All part numbers given are Radio Shack Part Numbers. Other parts of equal value may be substituted.

Parts List:

T1 - Transformer (273-1515) 18vct (2A)
V1 - 5 V Regulator (276-1770)
Heat Sink for V1 (276-1367)
D1, D2, D3 - 1N5400 Diodes (276-1101)
R1, R2, R3 - 10 ohm resistors (271-1301)
R4 - 100 ohm resistor (271-012)
R5 - Optional resistor (See text)
C1, C2 - 0.01uf capacitors (272-131)
C3 - 4700uf 35v capacitor (272-1022)
C4, C5 - 1uf capacitors (271-135)

If you can´t find a 7 pin DIN plug similar to the one on your Atari power supply, you can get a 5 pin DIN plug from Radio Shack (#274-003) for the power plug.

If you use the 5 pin plug, be extra careful as it is the same plug that goes into the monitor jack. If you plug 5 volts into the monitor jack, I am sure you would have problems, so paint the plug red or something like that if you use the 5 pin plug. Any of the terminals that are marked +5 or ground will power the computer. Be extra careful when hooking them up and observe proper polarity.

This power supply can be assembled without a PC board by using a 5 lug terminal strip. Be extremely cautious, as you will have 117 volts present. The two 117 volt wires on the transformer should be taped and insulated to prevent electrical shock. A PC board and plastic or metal case may be used if desired.

PRECAUTIONS

Some electronic experience is most essential to build this project. Don´t even attempt it if you don´t have the qualifying experience. Get a friend or someone knowledgeable in this area to help. Most "HAM" Radio operators would be able to help.

GETTING STARTED:

First wire the line cord to the two 117 volt power supply leads on the transformer and insulate them by taping, etc. You may also want to put a 1/2 amp fast-blow fuse inline on one side of the power cord for protection. You can tell which side of the transformer to hook the 117 volt line to because the low voltage side has 3 wires coming from it and the 117 volt side only has 2 wires.

Assemble the rest of the circuit as per the diagram. Look the circuit over very carefully before starting and if it is in a text file on disk I strongly recommend dumping it to the printer, or drawing it out on paper before starting the assembly.

If you use a metal case for your power supply, don´t let the regulator or heat sink get against it. Some people like to bolt the heat sink to the case for extra cooling but it will cause a short with this particular type of regulator. If extra cooling is needed bolt it to a piece of scrap aluminium and keep it insulated from the case. A plastic case really is better for our project.

At the top of the diagram you will find "D1" and "R5". This is the diode that allows the current to flow from the battery to the computer upon power failure. The resistor "R5" is a bypass "current limiting" resistor to allow the power supply to charge the battery. You may or may not want "Charger" capabilities. If you use a Motorcycle, or Car battery you will probably want to charge it a small amount. A 140 to 150 ohm, 1/2 watt resistor will allow about a 100 milliampere charge rate. A 700 ohm, 1/2 watt resistor would allow about a 20 milliampere charge. In order to determine other charge rates and the proper combination of resistors I suggest you purchase an Ohm´s Law Calculator from Radio Shack for the small price of about 50 cents. It is a small, sliding, cardboard chart similar to a slide rule.

A summary of the charging system: If you don´t want to attempt to build it just delete "R5" from the circuit and charge the battery manually, with a separate battery charger. You can insert a DC Amp Meter in series with "R5" to determine the amount of current that is actually going to the battery. The Diode "D1" will only let the current go in one direction which is from the battery to the computer. No voltage will be allowed to go "upstream" from the computer to the battery.

If you use lantern type batteries, they are not designed to be charged so delete "R5" from the circuit. I recommend a motorcycle, garden tractor, car, or gel cells be used, as these will keep the computer going for several hours. If you use gel cells be sure they are at least 5 Amp hour batteries. A 12 Volt battery is necessary, even though we are only ending up with 5 volts to the computer. The 7805 regulator needs at least 3 volts above it´s operating voltage to work properly. Thus the 12 volts, as 8 volt batteries are in short supply. Anything other than 12 volt batteries are not recommended, because that is what I designed the circuit to work with.

If you have trouble, questions, or suggestions please call me at my BBS in Springfield, MO (The Auctioneer BBS 417/887-4969 ,24 hours) or write to me at the address at the end of the text.

Please feel free to share these plans with your friends, or other BBS´s. They are intended for public distribution and may be shared or distributed freely.

Send all inquiries or donations to:
Pete Hunter, 2760 W. Whiteside,
Springfield, MO. 65807

Good Luck! Hope you enjoy the project! - sincerely, Pete Hunter.

The plans for the battery back-up are copyright 1986 by Pete Hunter Auctioneers Inc.

Opinion Poll

A.C.E.C. MEMBERS SURVEY

The club needs your feedback in order to develop a program of activities to help you get the most from your ATARI, whether for personal or for business use. In order for the officers to serve you better, we would like to know what programs, languages, etc. that you would like to see more attention given to, whether that means a few demos, a series of newsletter article of a Special Interest Groups.

Several categories are listed below; please feel free to add any additional comments as well. Please give your name and phone number if you wish to be contacted when a group is formed for a particular subject.

Please bring this questionare to the Feb., 1987 meeting, or mail to: Charles Lusco, 4624 Channing Terr. #C, Columbus, OH 43232

NAME ______________________________
PHONE ______________________________

LANGUAGES: (C, BASIC, LOGO, others)

____

WORD PROCESSORS: (AtariWriter, PaperClip, others)

____

SPREADSHEETS: (Visicalc, SynCalc, others)

____

DATA BASES: (SynFile, Data Perfect)

____

DOS SYSTEMS: (DOS 2/2.5, DOS 3, DOSXL, Sparta DOS, others)

____

FINANCIAL PROGRAMS: (Bookkeeper, Personal Finance, others)

____

SPECIAL DATA SYSTEMS: (TimeWise Astromony programs, etc.)

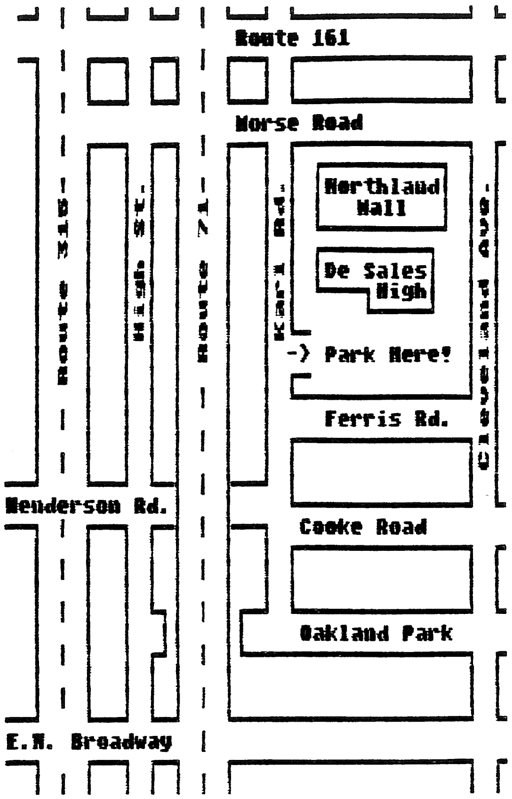
____

GAMES: (Zork, Ultima I-IV, Star Raiders)

____

OTHERS: (use the space below, and additional sheets as necessary)

[not to scale]



An official Users' Group, the Atari Computer Enthusiasts of Columbus meets on the SECOND MONDAY of each month. The meetings are held at 7:15 p.m., at De Sales High School on Karl Road. Meetings are open to the public, and consist of demonstrations and short tutorials of products for the Atari Home Computer Systems. Dues for ACEC are $12.00 per year, and include a subscription to Fuji Facts, and more!

MGL '87

Fuji Facts Newsletter
Warren Lieuallen, Editor
1652 Hess Boulevard
Columbus, OH 43212

TO:

MEETING: Mar. 9th, 7:15 pm