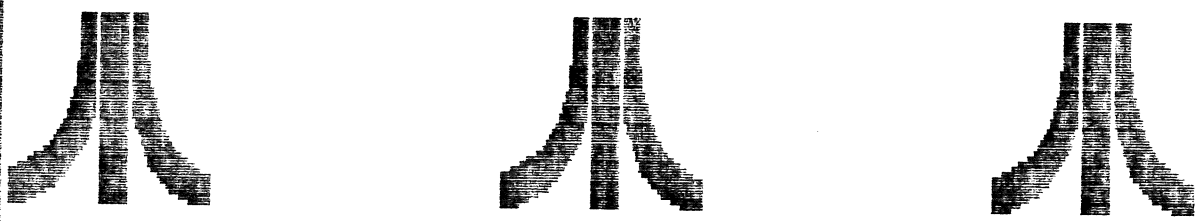
ATARU COMPUTER ENTHUSASISTS

OF COLUMBUS NEWSLETTER
SEPTEMBER 9, 1985 ISSUE 23
PRICE $2.00



Published by
Atari Computer Enthuasists of Columbus, Ohio

for ACE of Columbus membership. Dues are on an annual basis and entitle the members to all club benefits (Newsletter, Disk or Tape of the month, group discounts, etc.). Monthly meetings, at St. Francis De Sales High School, 4212 Karl Road, Columbus, Ohio are open to nonmembers. Meeting dates are the second Monday of the month at 7:30 pm. Disk of the Month distribution from 7:00 to 7:30.

PRESIDENT
Bill Eckert
6632 Lisamarie Road
Columbus, Ohio 43229
614-891-9785

VICE PRESIDENT
Joe Blue
6360 Sunderland Dr.
Columbus, Ohio 43229
614-436-7339

MEMBERSHIP CHAIRMAN
Tim Adcock
7544 Satterfield Road
Worthington, Ohio 43885
614-764-9492

TREASURER
Mike Compton
1342 Gumwood Dr.
Columbus, Ohio 43229
614-885-3757

DISK LIBRARIAN
Charles Lusco
4624 Channing Ter. Apt. C
Columbus, Ohio 43232
614-863-4016

PUBLICATION LIBRARIAN
Roger Stultz
2162 Eden Ave.
Columbus, Ohio 43224
614-471-5573

NEWSLETTER EDITOR
Norman Knapp
1222 Norton Avenue
Columbus, Ohio 43212
614-291-2849

ADDRESS ALL MAIL TO
ACE of Columbus
P.O. Box 849
Worthington, Ohio 43085

SECRETARY
Kathy Fellows
1719 Shaton Ct.
Worthington, Ohio 43085
614-889-4763

CASSETTE LIBRARIAN
Don Bowlin
230 Orchard Lane
Columbus, Ohio 43214
614-262-6945

PROGRAM CHAIRMAN
Don Noble
614-890-4333

ACE of Col BB SYSOP
Brent Borghese
614-268-0405

Machine Language: Putting It All Together

By Dr. Warren G. Lieuallen

Over the past few months, I have written a number of articles concerning the use of machine language, and the very closely related assembly language; I hope that I have helped in some small way to increase your interest in and understanding of this very complex subject. This last article will deal with perhaps the most practical use of assembly language to the "average" programmer--the Assembler/Editor.

When I explained that most machine language programming is actually done using the "mnemonic" assembly language op-codes, I mentioned that this would allow a program to be constructed that would convert these more "user-friendly" op-codes into the actual hexadecimal numbers that make up the machine language program, and that this program is called an assembler/editor. There are several commercially available assembler/editors for the Atari computer system. The most well-known are the Atari Assembler/Editor, by Atari (who else?!), and MAC/65, by OSS, Inc. As I have the former, that is what I will discuss. The MAC/65 is a more advanced (and more expensive!) assembler/editor, which is very highly recommended by assembly language programmers.

When using the assembler/editor program, a program is typed in very similarly to a BASIC program: line numbers are used, and commands follow according to the proper syntax; "remark" statements are allowed; variables may be defined and assigned values, and so on. Line numbers may be generated automatically, if desired, and renumbering of the program is also available as an option. More reminiscent of a word processor, a defined string may be searched for, and changed either globally or individually.

Once the program has been entered, it must then be "assembled". It is this stage which converts the assembly language program into the true machine language which the microprocessor will understand. Once assembled, the program may be immediately executed, or (more likely!) it may be debugged. As aids to fixing any errors, the program will allow you to trace through the program, examining the contents of all the registers, and watching the flow of the program through the memory locations. You may also step through the program, one command at a time, to exactly pinpoint the nasty "bug". By switching back to the editing mode, any errors may be corrected, and the program re-assembled. Finally, the perfected version may be saved to disk (or even to cassette tape), where it becomes just another binary file, to be used by yourself, and perhaps others!

Another way the assembler/editor can be used (and this is

my forte) is to load and subsequently modify a pre-existing machine language program. Whether simply personalizing the textual prompts, combining several subroutines, or adding a few additional features, this feature alone makes the assembler/editor a valuable asset. Programs which are presently satisfactory, but not quite perfect as far as you are concerned can now be upgraded to your own high standards of performance! If nothing else, it will at least allow you to discover just how someone was able to make your Atari do a certain special something in machine language, thereby increasing your understanding of computer programming, and allowing you to incorporate another item into your "bag of tricks"! And as you know, every little bit helps!

I hope you have enjoyed this last look at the world according to a 6502 microprocessor! If you come across any particularly elegant machine language solutions, I would certainly appreciate hearing about them.

==========

ATARI AID

by Dr. Warren G. Lieuallen

For a fabulous scrolling screen effect, try adding this line to the start of a program...

```
10 POKE 622,255:GRAPHICS 0  (Any other graphics mode with a text window should work also.)

20 POKE 731,255:POKE 730,1:POKE 752,1  (This last POKE just removes the cursor.)
```

The effect should last until you change graphics modes. To change modes and keep the effect, just repeat these lines with the desired mode.

This routine appeared in a program called PHONE.LST (written by Dusty Moss), which appeared on the ACEC bulletin board.

==========

ST STATUS: September 1985
by Norman Knapp

It's now been several months since the developement Atari ST systems were delivered to software producers and several weeks since the release of the monochrome and color versions to the microcomputer market place. What then is the status of the ST in microcomputer community.

Let's take a brief look at how Atari's ST arrived its current status. An initial enthusasism was generated because of user friendliness of GEM, the power of the 68000 microprocessor, and its low list price. Using baseball terminology, the ST was first released at exhoribant prices to software producers, few of whom took up Atari's offer: STRIKE 1. As a result of this initial blunder, there are very few companies producing software for the ST. Most of the programs advertised are not available for immediate shipment. It seems that software houses are waiting to see how well the ST sells.

What's out there for the ST besides the book published by Abacus Software, "Presenting the ATARI ST," which I reviewed in a recent issue of our newsletter. An article in the September issue of Antic lists the following companies as ST developers: Atari Corp., Haba Systems, Batteries Included, Rising Star Industries, Rugby Circle, Dragon Group, Abacus Software, Datasoft, Electronic Arts, Sublogic Communications Corp., Accolade/F.T.L. Software, Activenture, and Sierra On-Line. A company not listed here, VIP Technologies, placed an ad in the same issue for an ST spreadsheet, VIP Professional, (a 123 clone). Only software by produced by Haba was advertised in the same issue of Antic: a checkminder, a communications package, Hippo C, a wordprocessor, and a spreadsheet. A phone call to a local dealer revealed that Zork I and Wishbringer, text adventure games by Infocom, were available for the ST, as well as Express, a word processor.

Computer magazines, Antic and Analog, are publishing a few articles about the ST. ANTIC has set aside a section for the ST; highlights of recent articles have dealt with 800/ST file transfer using Kermit, and a performance test. A few short programs have been published in both ANTIC and ANALOG, but have not been useable since most of programs were not written in Logo, the language shipped with the early machines. File transfers are not available for early purchasers since the ST version of Kermit is unavailable.

When ST was released in July, Logo was the only language that came with it: STRIKE 2!!!!!! There are two reasons for this conclusion: the universal acceptance of Basic as the first language for a microcomputer and the slowness of Logo. Two years ago, an article was published in Byte (Nov 1982) which demonstrated the beauty and simplicity of Logo programs compared to the corresponding PL/1 program. The program was keyed in almost exactly as published and after a short period of debugging during which I was able to move readily between workspace, editor, and running of the Logo program, the program was up and running, drawing a complex pattern of pentagonal spirals. Only one problem, during the time required to make a picture, I could have gone out to the DQ for a milkshake. The Logo turtle can not move too fast, otherwise the educational value of watching him draw his

patterns would be lost.

Whether Atari makes a hit with the ST or strikes out, only time will tell. Delivery of applications and programming software are scheduled for October, which is not so very far away. It should be noted that the problems Atari is having with the ST are the same as those experienced by other manufacturers when introducing an entirely new microcomputer product.

Recent reviews in Compute and Creative Computing have been quite favorable, desribing it as the best microcomputer for the money.

****

ST Tip: Telecommunications

The ST's terminal emulator program, VT52 Emulator, works nicely with the Hayes Smartmodem connected to the ST's RS232 port. However, I've been told that a telecommunications program is required for downloading files.

by Norman Knapp

****

ST Tip: CP/M-86

The ST's operating system is Digital Research's CP/M-86, a 16 bit version of CP/M. When an applications program is in use, the user can exit from the GEM shell and issue CP/M commands directly from the A> prompt. The following book will help the ST owner to become familiar with CP/M-86:

David E. Cortesi, "INSIDE CP/M-86, A Guide for Users", Holt Rhinehart and Winston, New York, 1984.

by Norman Knapp

****

PRINTSHOP Tip: Drawing

Drawing Printshop graphics can be done with the joystick, KoalaPad, or Atari Touch Tablet. The Super Sketch device can also be used for drawing Printshop graphics. The advantage of Super Sketch is that drawings on 8.5 in wide paper may be copied. Due to differences in coordinate conventions, the picture must be placed upside down on the Super Sketch surface. Unfortunately, its manufacturer, Personal Peripherals, is no longer in business.

by Norman Knapp

****

Computer Furniture Tip

Plans for making a "roll-top computer center" are available from the Columbus Dispatch, Craft Patterns, 2200 Dean St., St. Charles, Ill. 60174. Ask for Craft Pattern C0462 and enclose $3.95.

Our librarian will have a picture of the finished project on file.

by Norman Knapp

CEditor's Column

by Norman Knapp

This is the issue of the ACE of Columbus Newsletter that I will be editing. Ir's been an interesting year and a half that I've held this position, but we need someone new to carry one the job.

I would like to thank all the members who have contributed items for publication, especially our most prolific writers, Charles Brown and Dr. Warren Lieuallen. There is just one loose end I am cleaning up in this issue; my apologies to Charles Brown for missing the last page of his article in the August issue. That article is reprinted in entirity in this issue.

In this issue I am publishing something unique for computer club newsletter, a piece of fiction, "Blackfire", by Krista Knapp, my daughter. I hope you enjoy it.

USING THE ATARI ASSEMBLER EDITOR PART 3

In the past I have tried to explain to you how machine language programing works.The best way that I know how is to give you an example.

In Basic language if you want to print something to the screen, you would use the Print command like this:

```
Print "HELLO ATARI USERS"
```

Anything that you put in between the quotation marks would be printed. It could be any combination of letters or numbers or special characters. It also does not matter how long it is.

I have listed below an actual machine language program. This program does the same thing as the one line of Basic code listed above:

```
1       2       3                       4                   5
0000            10              *=  $3000
3000 A200       20              LDX #0              ;CHANNEL OFFSET
3002 A909       30              LDA #9              ;WRITE command
3004 9D4203     40              STA $342,X          ;ICCOM
3007 A921       50              LDA #MESS&255       ;LOW BYTE OF MESSAGE
3009 9D4403     60              STA $344,X          ;ICBAL
300C A930       70              LDA #MESS/256       ;HIGH BYTE
300E 9D4503     80              STA $345,X          ;ICBAH
3011 A911       90              LDA #17             ;LENGTH OF MESSAGE
3013 9D4803     0100             STA $348,X          ;ICBLL
3016 A900       0110             LDA #0              ;H.B. OF LENGTH
3018 9D4903     0120             STA $349,X          ;ICBLH
301B 2056E4     0130             JSR $E456           ;CIOV
301E 4C1E30     0140 STOP        JMP STOP
3021 48         0150 MESS        y.BYTE "HELLO ATARI USERS"
3022 45
3023 4C
3024 4C
3025 4F
3026 20
3027 41
3028 54
3029 41
302A 52
302B 49
302C 20
302D 55
302E 53
302F 45
3030 52
3031 53
0 ERRORS
```

As you can see you have a lot more than one line to type. In fact you have 15 lines to type in order to do the same thing. Also you have to be more specific in what you are doing.

Whenever you put or get something to or from another device besides the computer, you have to use the I/O control blocks. Other devices include the disk drive, program recorder, printer, TV screen or monitor, modem, keyboard, or the RS232 ports. These blocks are like channels that data goes back and forth on. You must be able to understand them in order to better understand programming. When you print to the screen you automatically use I/O block 0. This channel is always open for your use. If you do a Lprint to the printer (in Basic) the computer will automatically open channel 7, do the printing and then close the channel. In machine language you would have to go through all the steps. You would first have to to open the channel. Then send your data over it. Then close the channel. You can easily see that Basic is much simpler.

The listing shown above is from an assembled program. It shows both the source code (the code typed in by the programmer) and the resultant object code created by the computer). In column #1 you see the memory location for that line of code. In column #2 you see the hexadecimal value for the object code of that line. In column #3 you see the source code line number. In column #4 you see the actual source code command. Finally in column #5 you see the comments that explain what is going on. The semicolons that you see in the beginning of column #5 are the same as a Rem statement in Basic.

Now I will try to explain the program line by line. In line #10 I am telling the computer to store the object code in 3000 hex. This location would be 12288 in decimal. You have to know where to store the code. You could write over some important data and crash the computer. In line #20 I load the x register with the number 0. The x register is used as a an offset. By using #0 I am telling the computer to go that I want to use I/O channel 0, so the computer will know I want to print to the screen. This channel is always open, so I won't have to go through the trouble of opening it. In line #30 I am loading the accumulator with the #9, the number for the write command. In line #40 I store the contents of the accumulator into location 342 hex(834 decimal). This is called the iccom where the computer will find the command for the type of action to be taken on this channel. Since we put the 9 there, the computer will know it is supposed to do a write. In line 50 I load the accumulator with the low byte of the message to be printed. Then in line 60 I store the contents of the accumulator into location 344 hex(836 decimal). This is called the icbal. It is the low byte of the buffer address for the data that is to be transferred. In line #70 I store the high byte of my message in the accumulator. In line #80 I store the contents of the accumulator in location 345 hex(837 decimal). This is the icbah the high byte of the data buffer address. In line #90 I load the accumulator with the number #17. This is the number of characters in my message. You have to know exactly how big your message is. If your number is too small then your whole message won't be printed. If you have too big of a number then you will have a bunch of garbage at the end of your message. In line #100 I store the contents of the accumulator into location 348 hex(840 decimal). This is

called the icbll the low byte of the buffer length for the data to be transferred. In line #110 I store the number 0 in the accumulator. This is the high byte of the number of characters in my message. If my message was more than 256 characters then I would have to put a 1 here. In line #120 I store the contents of the accumulator in location 349 hex(841 decimal). This is called the icblh). It is where the high byte for the length of message would be stored. Since I am putting a 0 here the computer knows that the message is less than 256 characters. In line #130 I do a jump to subroutine E456 hex(58454 decimal). This is the same as a Gosub in Basic. In this program we are going to a subroutine that is already built into the computer ROM. It is called the ciov. This is the central input/output(CIO)utility entry. This utility handles all the I/O operations or data transfers. Anytime you do anything with any of the I/O channels you will be using this built in routine. No matter what language you are using. In line #140 you have a label called stop. Then I have a command that says jump to a label called stop. This is the same as a line going to itself in Basic. Finally in line #150 I tell the computer what mess is. Using the dot byte command I define what the variable mess is. This is so the computer knows what to print. It would be like saying mess$="hello atari users" in Basic. Then just printing mess$.

Right below line #150 you see 2 columns of numbers. The 1st column shows the memory location for each letter in my variable labeled mess. Then the 2nd column shows the hexadecimal value for the ATASCII code of each letter in the variable mess.

I have tried to take you through step by step and show you what it is like. In this one example 1 line of Basic code is broken down into 15 lines in assembly language. As you could see the accumulator got some very heavy use in this short program. Just think how many times it gets used in a major one. Every time you want to put something somewhere you most likely have to put it in the accumulator first.

I hope that you have a better understanding of assembly language by now. If you have any questions please feel free to ask me. Even though I am only a beginner. I will be glad to try and help you out. I might not know the answer but I will try to find it for you. Please don't be scared of all of this assembly stuff. As I said before, if I can do it then anybody can.

by CHARLES W. BROWN

Blackfire
by Krista Knapp

I said from the beginning it was a bad idea.Not that I'm complaining , it's just that some people never listen to good advice. A party of quality adventurers shouldn't risk their skins to kill a common dragon. Well, Blackfire wasn't a common dragon but that's beside the point. The point is we were in the middle of the Pendaris Mountain Range; one of the worst storms I'd ever seen was brewing, and we just lost Randolph who had been our only cleric.

"Radd, get over here!"

That's me."I'll be right there," I yelled back.

It wasn't much. We'd just come to a forking in the trail, and , because I was the ranger, they wanted me to decide which path to take. I checked the trails for tracks and although I found no telltale signs of a dragon there were plenty of recent goblin tracks. Since it wouldn't do to unduly upset anyone, I didn't mention them. Eventually we took the northern trail because the wind was starting to blow real bad and it looked like we might find shelter in that direction.

We were walking in standard marching order. It went me; Fern , the monk; Tavnish, a half-elven fighter; Evro, an elven magic-user; Treddo, the thief; Trik, our illusionist; and Camor, a dwarven fighter. Camor had some objections to bringing up the rear but the rest of us convinced him it wasn't that dangerous even though it was. You see, the rear was most often attacked first. I didn't mind leading because this was obviously goblin country and goblins never used frontal assaults.

It was almost an hour later when the ground started sloping steeply upward and I guessed that the valley we had been in was ending. I was becoming uneasy for this meant our chances of finding shelter before the storm came would be greatly reduced and on top of that for the last 15 minutes I had the strangest feeling that someone was following us. But as everyone else was busy complaining or arguing I didn't bother to share my suspicions with anyone. Just as I was thinking this I heard a loud rumbling sound come from above that had nothing to do with the storm.

"Scatter!" I cried as I realized that an avalanche of boulders was thundering down on us. Then, thoughts of all else was driven from my mind as I tried to find a safe place to go to. Fortunately for us this wasn't a barren slope like some and it had some tall trees that we could climb.I'd just reached the first branches of my tree when a rock struck me in the forehead and I lost conscious.

When I finally started to wake up it was because someone was yelling.

"Radd, Tavnish, Camor, Trik, Evro, Fern! Where are you?" It was Treddo. I wished he would stop, the loud noise was making my head spin.

"I'm over here," I replied weakly. Apparently he heard me because he stopped yelling and started making his way over to where I was. By the time Treddo was next to me I had pulled myself up and was looking around me. Almost immediately I was sorry I did that for the landscape was covered with rocks and it was easy to see that Evro and Tavnish hadn't made it to a tree and had decided to try and outrun them. Of course they had failed. But I didn't have much time to think about them because everyone else was coming to and starting to ask questions.

"Who's still here? Is everyone all right? What started that avalanche? " I finally cut through the babble of voices to get order restored.

"Evro and Tavnish are dead. I think the rest of us are all right and if we want to keep it that way then we'd better get moving,"I said. To my disgust Fern started questioning me.

"What do you mean 'if we want to keep it that way then we'd better get moving?' Get moving from what? Do you know something we don't know?" he asked,his voice dangerously smooth and even. Fern was staring at me and getting angry. I realized it had to be the avalanche that made him that had way and I decided to placate him, after all it wouldn't do to have them doubting my expert leadership. But I was shocked nonetheless. He was acting like I didn't share everything I knew with the party!

"Of course I've told you everything I know Fern! Why would I withold anything from you? I only meant that there might be another avalanche and that the storm cloud is almost overhead now. But if you really want to risk this slope again and not get going on a sidetrail then why don't we just stand here and keep arguing?" I retorted.

Fern murmured something about he had only asked a question and he didn't mean that he wanted us to risk the slope again and of course he wanted to get moving. I was satisfied and we started up the sidetrail I mentioned. When we reached the height at which the avalanche had started I called for a rest in spite of the rising wind and then went off to investigate the site of the start of the avalanche. Sure enough the rocks had been loosened and there were goblin tracks about. Since the incident had already taken place I saw no reason to tell the others that it hadn't been an

accident. I wrapped my cloak about me and rejoined them. They were having an argument about how close Blackfire's lair was and how they were going to kill him. Camor was bragging about his magic sword when I interupted him after I noticed that Fern wasn't with them. Trik spoke up and said that he decided to scout around upahead. I was livid. Scouting was dangerous and he hadn't even cleared it with me first! I got everyone going again although there were some grumbles, it was starting to rain and Trik was in favor of stopping. I had almost succumbed to his suggestions when Fern appeared on the trail ahead of us.

"Over here!" he called,"I think I've found a decent cave.I'm going to check it out.Follow me!"

"Don't you dare check it out until I've checked it for traps!" Treddo roared." I'm the thief and you're just a monk"

"Get back here," I added but I knew he wouldn't listen to me. No one seemed to listen when I told them about the dangers of mountain caves and like things. This was one thing I couldn't understand about them. Fern, of course,didn't listen to us and ran right into the cave. We stopped dead in our tracks when we heard a thunk and a low strangled cry. Fearing the worst, I cautiously walked up to the cave and looked in. Fern was lying dead on the floor with a crossbow bolt in his throat. It had come from a trap that had been set there not very long ago. We dragged his body over to a niche in the cave wall and blocked the opening up. Secretly I was glad he was dead. He had been a good monk and good monks were few and far between but lately he'd just been asking too many questions. It was better that he died this way then by my sword.

Treddo checked the cave for more traps but there were none. However, he did find an entrance to a secret tunnel at the back of the cave. When we opened the door the heat in the tunnel rushed out at us and we all had to leap back a couple of steps. On closer inspection the tunnel had scorch marks on its walls and it smelled of sulfer. We decided it was an air vent to Blackfire's lair. It was just what we'd been looking for! All thoughts of resting left our minds and we started down the tunnel. Treddo went first, then Camor, then Trik, and lastly me. As we went down the tunnel the temperature steadily rose and by the time we reached the end it was about 130 degrees. The tunnel ended in a huge room. We didn't look at all the gold, silver, gems, coins, jewelry, crowns, or scepters. We stared at Blackfire. And stared. And stared. Looking back I know we lucky that he was asleep or he would have been able to kill us right then and there. Normally we weren't this slow but Blackfire was a sight to see. He had to be at least 70 feet long and his hide was the color of midnight. He seemed to glow and shine and there was a certain aura about him. I didn't know how we expected to kill him. It

always seemed so easy when we discussed it before but now my mind was a blank. At that moment Treddo noticed a diamond twice the size of a man's fist laying in between Blackfire's forelegs. It transfixed him.

"Just like the Arkinstone," he murmured. Then,"I've got to have it," he breathed and started creeping across the room.

"Wait," Camor called softly, "We have to kill him first!" But Treddo was deaf to our pleas and he moved up close to Blackfire. I don't know if he woke Blackfire up or if Blackfire had been playing with us all along but at any rate he woke up right then and with a mighty roar he swepted his claw down and tore Treddo in two. At this Camor bellowed his dwarven battlecry and sank his magic sword into Blackfire's neck. It affected him about as much as a mosquito bite would. Trik took one look at this and screamed "Retreat!" while racing for the tunnel. I thought it sounded like a good idea but I wasn't about to leave without any treasure and so I grabbed the closest thing which happened to be a golden ring and fled for the tunnel. Camor abandoned his sword and followed close behind me.

We would have all made it if Blackfire hadn't lived up to his name and breathed fire after us. Camor bore the brunt of it and was incinerated on the spot. My back was scorched and I still have the marks to prove it but other than that I was alright. Trik was out of the tunnel and into the cave by the time I'd reached the end of the tunnel. Therefore he had no time or place to hide from the waiting goblins while I could duck back down the tunnel a bit and hide. Trik tried to cast a spell but they put an arrow through him before he could finish.

When Trik died I thought I was dead but just then I remembered the ring that I'd picked up. It was glowing and felt heavy. I realized it was magical and I tried it on. Wonder of wonders it was a ring of invisibility! I naturally took advantage of it and just walked out through the goblins. My trail was wiped out behind me by the storm which had finally broken and was raging. I couldn't believe my good luck!

My only regret was that the party had died in the attempt. "Oh well," I thought,"It isn't as if it was my fault. After all, hadn't I said from the beginning that it was a bad idea?"

DR. WARREN G. LIEUALLEN
12/9/85

Sept. 14th Sat 10-6

Both stores open at 9 for

Atari Sale

Software only

min. 60% off

up to 90% off

Sept 19th SIG meet.

Byte – Septem

800XL to 256K

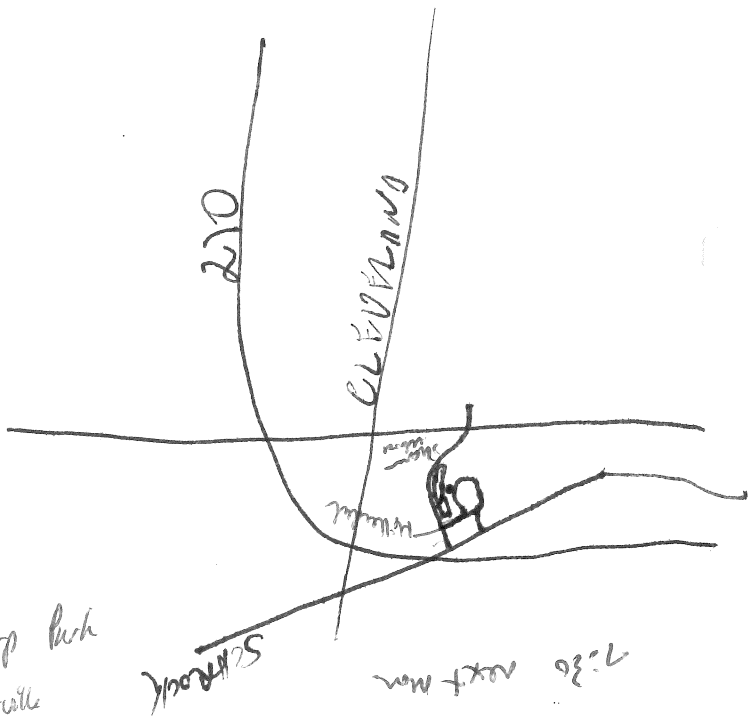
~ $35

Hamilton Ave.

Brooks Edge Corp Park

in Wyoming

near Penney's



68000 kit for 68

ACE OF COLUMBUS NEWSLETTER
P.O. BOX 849
WORTHINGTON, OHIO 43085