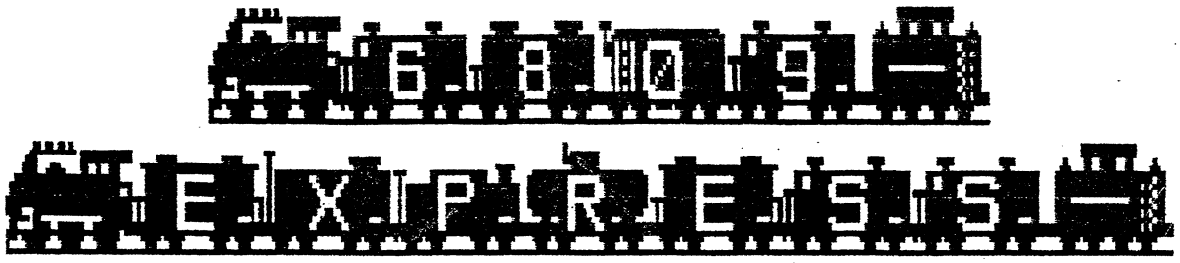


The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

December 1989

BARGAINS

by Clyde Gano

We will be collecting 1990 Club dues at the Christmas Party Meeting on Friday, December 29. As in the past, there is no increase- $12.00 for the year for the entire family. What a bargain!

Among the many benefits of the Club, we still continue to order **RAINBOW** subscriptions. The Club discount is $2.00 for each subscription. If the subscription is a renewal, please bring a mailing label.

Ordering as a group gives us the advantage of quantity discounts as well as a saving in shipping costs. If you need anything from **MEI/MICRO CENTER**, see your treasurer. If we can put together a $25.00 order, we will be buying colored ribbons from **RAMCO**. They seem to have a good assortment available. We welcome any suggestions as to supply sources that will benefit the Club.

MINUTES OF THE NOVEMBER 24, 1989 MEETING

Vice-President **Rick Hengeveld** opened the meeting. The minutes were approved as written in the 6809 Express. **Clyde Gano** read the Treasurer's Report which was also approved.

OLD BUSINESS:
Questions were answered concerning the disks ordered from MEI: double-sided, double-density. There are two disk cases for **Danny** and **Adam**. If they are not interested in them, the boxes will be offered for sale in January. The Phillipsburg Mall show possibility is on hold at the present time. The Christmas Party will be held at the December meeting.

NEW BUSINESS:
Nominations were closed and voting for the new officers proceeded. The new officers were elected unanimously.

(continued on page 2)

(MINUTES continued)

President	-**Rick Hengeveld**
Vice-President	-**Arthur Spengler**
Treasurer	-**Clyde Gano**
Secretary	-**Ilene Spengler**
Editor	-**Peter Unks**
Librarian	-**Bruce Navarre**
Publicity	-**Ronnie DeGarmo**

Art reported on a new program from our friend in Australia. It prints borders for the booklet and also for sheets of paper. Members were reminded to send newsletter information to **Peter Unks** as early as possible. The meeting was adjourned and Random Access followed.

The program for the evening was presented by **Rick Hengeveld**. It involved showing various graphics and a little insight into OS-9. The information was enjoyed by all.

Submitted by,
Mary A. Brown
Secretary

GAMES AND WINDOWS

A REVIEW

by Rick Hengeveld

Kings Quest is an adventure game that mixes the use of the keyboard with a joystick. This game is both challenging and pleasing to the eye. That along with the fact that this game never seems to play quite the same twice in a row makes it a winner! There are some problems. First the program is large enough to require 512K and at the price of RAM chips these days that could put a lot of people out of the Kings Quest market. The other thorn is the constant disk switching that's required. Kings Quest works great on a hard drive, But when you have to switch between the 5, yes 5 double sided flippys, the action slows down. Still I can't fault Sierra for giving us such a large program, after all it makes for a super game! I've seen Kings Quest on both the IBM and the Amiga and I can assure you that the Coco version is every bit as good! It is available from Tandy for about $35.00.

Now, a look at Window Master from Cer-Comp. If you're a basic programmer this program should really be near the top of your list of gotta-haves! What Window Master does is to modify RS-DOS to allow new commands to create windows, Icons, Pull-down menus and a whole lot of other goodies. All these are the available right out of Basic. Multiple windows (31) can then be opened and moved on the screen. I've been looking for a lot of these features on my Coco for quite sometime. Now with Window Master, I have them, And all from good old RS-DOS! So far I

(continued on page 3)

(WINDOWS continued from page 2)

haven't found any bugs in the program itself. However I must tell you that the documentation leaves a lot to be desired. Therefore if your not well versed in the use of Basic or if your just beginning to write your own Basic programs, I'd recommend you think carefully about using Window Master.

Window Master takes no memory away from basic. It resides in the upper RAM, so it is not compatible with any machine language programs. Cer-Comp currently has about 5 programs that run in the window environment. Cer-Comp hopes to come out with many more of these point and click type programs. Window Master comes with both a 128K and 512K versions, However to fully use Window Master 512K is highly recommended also you'll need a Tandy High Res interface and a mouse. While all but a few of the RS-DOS commands work with Window Master (some low-res commands are disabled).

I wonder if this operating environment will take off. For example, I'm currently writing a disk utility that will run under Windows. But for anyone else to use it they must first own a copy of Window Master. The codes and tokens that Window Master generates are not recognized by RS-DOS. Therefore the non-programmer will have to see quite a few programs on the market that run under Window Master before he or she decides to invest the $70.00 in Window Master itself.

Along with the ability to run windowed programs you also get a full 68 Granule Ram disk and the elimination of most of the known DOS bugs that were found in the Coco, such as the DISKINI bug.

This program is a winner! I just hope to see enough applications of this program to make it a commercial success.

6809 EXPRESS

The "6809 EXPRESS" is the official publication of the PENN-JERSEY COLOR COMPUTER CLUB. The club is based in the greater Lehigh Valley of Northeastern Pennsylvania including sections of Northwestern New Jersey. Any non-profit organization may reprint any part of this newsletter provided credit is given. PJCCC will gladly exchange newsletters with any other computer club. Send requests to EDITOR, 6809 EXPRESS, Penn-Jersey Color Computer Club, P.O. Box 8242, Lehigh Valley, PA, 18001 PJCCC assumes no responsibility for errors or omissions. PJCCC assumes no liability for damages resulting from the use of any information or programs contained in this newsletter.

President Roni DeBarmo (815)837-1571
Vice Pres. Rick Hengeveld (815)767-9344
Editor Reinhold Radke (801)859-6161
Secretary Mary Brown (815)797-7744
Treasurer Clyde Gano (815)536-8386
Librarian Eric Rhyder (815)377-6185
Publicity H. Peter Unks (801)454-5015
Bulletin Board Jim Mangan ((815)866-1805
West Coast Correspondent
Tony Dappelini (818)243-6899

The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

H. Peter Unks Acting Editor
145 Seventh Street, Morris Park
Phillipsburg, NJ 08865

FIRST CLASS MAIL

6809 EXPRESS

The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

MAVERICK BBS

215-760-0456

NOVEMBER 1991

THE 6809 EXPRESS RIDES AGAIN!

by Peter Unks

After an absence of several months because of technical problems, the **6809 EXPRESS** is back again.

The problems had to do with loading the copy (which was downloaded from **The Maverick**) into the editor's Max-10 word processor. Files would load, but would be inexplicably cut off somewhere before the real end of the text. Very distressing! I am happy to report that the difficulty has been overcome.

How, you might ask, was this accomplished?

Seeking a solution involved calling the creators of Max-10. They suggested noise in the transmission contaminating the files. I disconnected every cable, checked them out with great care, then reconnected the computer to the modem and phone line.

That failed to solve the problem.

A Random Access session at last month's meeting offered better solutions. Some bit of code might be hiding in the "packing" of the Xmodem protocol tricking Max-10 into a backspace. I examined the disk with a disk editor and rewrote the appropriate sectors back to the disk minus the "packing". That didn't help either.

I gave up on Max-10 and chain loaded all the files into VIP Writer just as they came down from the bulletin board. Then I saved the whole file to disk planning to finish the newsletter the next day.

As an afterthought I attempted to load this huge file into Max-10 to see what would happen. It worked! I tried this before with individual files with no success. Brains are no substitute for dumb luck!

The Maverick Report

Rick Hengeveld
9/2/91

The Maverick continues to grow with several out of state users who have somehow managed to get the Maverick's number! Nearly 200 messages have been posted. And well over 300 callers have been logged to the system! The system currently supports 31 users. We're starting to see an influx of users from the Philadelphia area using the Maverick. So our growth is slow but steady! Those who logon, be sure to scan the download section. There are a few new goodies that have been uploaded.

TREASURER'S REPORT

By Clyde Gano

Statement date 7/28/91	
Bal. on hand 6/28/91	$215.53
Receipts-	
Sale- 18 discs	9.00
destroyed ck#249 Bell PA	8.42
Total Receipts	$232.95
Disbursements-	
Hillcrest Shop 6809 EX June, July/Aug. CK#250	$19.20
Bell of PA June & July CK#251	16.79
Total Disbursed	$ 35.99
Balance on hand 7/28/91	$196.96

****

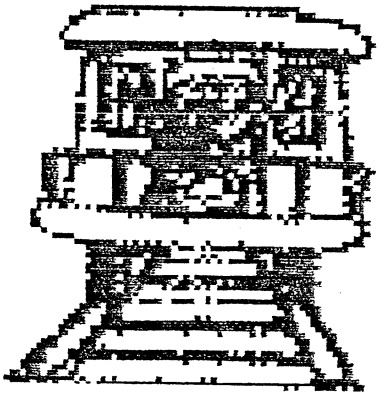
THE PRESIDENT'S REPORT

Rick Hengeveld

It's good to see the 6809 back in print! Summer is slowly coming to an end and it will be back to the keyboards for some of us. Soon it will be time for the membership to start to plan next years activities. So ask yourselves what you'd like to see at the upcoming meetings. We still have to look at compilers this year. Be sure to attend the upcoming meetings as I'm sure you'll find them very interesting.

****

PLAN TO ATTEND THE
ANNUAL CHRISTMAS PARTY!
DECEMBER 27 1991
BRING SOME GOODIES
TO SHARE IF YOU CAN!



THE LIBRARY CAR

AL WAGNER

Welcome once again to the PJCCC library car. I trust you've had a good summer. Pull up a couple of crates over here by the potbelly stove I've lit to chase the autumn evening chill. This month my planned discussion will take a side track for a bit and we'll discuss a question that came up at the meeting.

Pete asked about windows and window types. The OS9 that runs on the Color Computer has eight device windows and six different types of windows. A device window is a device just like a disk drive, a printer, an RS-232 port, or any other piece of hardware you could connect. OS9 deals with it in the same way. Device windows have one driver for all the windows and a different device descriptor for each window.

Seven of the eight windows must be created by software. The eighth, actually the first, is different as it is the green screen that is created by the boot and is the screen in which the machine naturally comes up. The device name for this window is /term. (Note that in OS9 all device names are preceded by a slash.) The other seven windows are /w1 thru /w7. Each of these last seven windows can be brought up in any of six different window types. Pete's confusion, I believe, came from the fact that each of the seven device windows has a default type and he had confused the type with the device number. Its very easy to do when one is just starting out in OS9. The default window types are listed in the OS9 manual in the section on windows on page 1-3. Again these are default types and any window, other than term, can be set to any type.

Pete referred to a particular application program as requiring to be run from a type 8 window. How come I've said that there are only six different types? Microware found it necessary to skip a couple of numbers. I know, you want to know why. Well, there are mysteries that may never be unraveled.

(go to next page)

The list of device window types are listed, again in the windowing section of the OS9 manual, on page 3-13. Type 8 is a graphics window 320 x 192 pixels capable of sixteen colors and requires 32K of memory. That blue screen you may have seen me use at the meetings, is a type 2. It is a text only window 80 x 24 characters capable of eight colors and requires only 4K of memory.

There are two commands to bring a device window to life. (Anything to confuse the beginner.) They are Wcreate and Display. For the most part you can choose your poison as they really do the same thing. The difference is that Wcreate use decimal numbers and Display uses hexadecimal numbers.

Display can be easier to use as the command modifiers are given in hex in the manual. But you will have to convert the size and location of the window from decimal to hex. To use Wcreate, you can use the size and location quite readily, but you must convert the modifiers. I will work with Display as it is the one I got started with, but realize that Wcreate will do the same thing except for hex to decimal conversion and a slight difference in the sequence the modifiers occupy after the command.

First one must use the INIZ /w# command to establish the window. Then one of the above commands is used. The command I use to create a graphics window eighty columns of text wide by twenty-four rows high is as follows (yeah that's right, text! OS9 can display graphics AND text on the same window IF you merge a font file with the device once you create it.): display 1b

20 07 0 0 50 18 0 1 1 >/w1 [cr]. (The brackets are to show the carriage return key rather than the actual letters "c" and "r".) The "1b 20" is the hex code for the command modifier "device window set". The "07" is the window or screen type. In this case, type 7. The "0 0" sets the upper left corner of the window to the upper left corner of the monitor's screen. The "50 18" sets the size of the window, width and height respectively. The "0 1 1" set the foreground, background, and border respectively. Because this is executed in my startup file which executes on the /term window, I must redirect the command to /w1 where I want the action to take place. The redirection takes place courtesy of the > symbol.

In addition to the above device windows there are overlay windows. An overlay window is just as its name implies, a window that lays over existing information on the screen. Where device windows cannot be laid over each other, overlay windows can be laid over anything as long as there is a device window at the bottom of the pile. You can only write to the

topmost window in the pile.

When you open an overlay window, you have the choice of saving the data under the overlay window or not. If you choose to save it, the original data will be written to the screen at the termination of the overlay window. The decision to save or not to save is not as simple as it seems. If you save the data it must be rewritten to a memory location other than that for the screen (this is automatic and you need not be concerned with implementing the rewrite). If your program already has memory limitations stretched to the max, you may wish to recreate the data rather than consume the memory.

I hope those of you curious enough to want to know more on windows have access to The Complete Rainbow Guide to OS9 Level II or another "third party" book on OS9. I will cover windows again a little later on in this series as some other commands need to be gone over before we are really ready to go further. Next month I intend to pick up where I left off in July. I see we are pulling into our station once again. Have a good month and I'll see you at the meeting. If you have any questions that need answering sooner, feel free to stop up at the Maverick BBS and drop me a note. I get up there several times a week to pick up my mail, so I should be able reply fairly quick. Happy computing!

****

DON'T MISS THE NOVEMBER 29 MEETING!

BASICALLY SPEAKING
Rick Hengeveld

I've been reading Al Wagner's articles in the 6809 dealing with OS9 since the Express has been resurrected, and found them very informative. I thought that it would be nice to have an RS-DOS counter part for those that don't speak in OS9. This will be my first attempt, While I certainly don't claim to be an RS-DOS guru, I will try to smooth some of the rough spots for some users.

Since this is the first of this series I intend to keep the subject matter as simple as possible. So we'll discuss the most used disk commands.

BACKUP (Syntax) BACKUP 0TO1 or BACKUP0 for single drive operation.

This command will copy an entire disk to a another disk. If any files are present on the target disk they will be erased.

Also Backup may not copy some disks due to various copyguard schemes. In such cases you may use the Spit'n'Image utility

(go to next page)

to copy the disk. While this copy program is not effective against some of the newest copyguards, it is worth a try.

COPY (Syntax)
COPY"FILENAME/EXT:0"TO"FILENAME/EXT:1"
This command will copy a single file to another disk. Some notes on the COPY command; First you must use the complete file name including the extension for the command to work also note the placement of the quotation marks. You may decide to alter the file name during the copy procedure, ie.
COPY"TEST/EXT:0"TO"RENAME/EXT:1" In this case you'll have the file TEST copied to drive 1, under a new name. One further item according to your RS-DOS manual the copy command can't be used on a single drive system, this is not true! If you assign drive 0 as both your source and target drives you may make a copy on a single drive system.

NOTE: Both Backup and Copy commands erase anything in the systems memory.

While it is possible to use a utility program to execute these two functions it is quicker and easier to simply issue these commands from basic if there is only one or two copies or backups to be done. When loading programs from disk you may notice your Coco drives load only a portion of the data and then the drive heads jump around until the rest of the program is found. This is due to file fragmentation. ie. The file is not in one solid block on the disk, it is instead fractured into at least 2 blocks on different areas of the disk. While this won't affect the programs operation it will take longer to load and cause more wear on your drive. Using the backup command on a disk like this will not alter the fragmentation of the file, however if you use the copy command and move the file to a fresh disk the file will be transferred to a single block thereby making the file faster loading and reducing drive wear. Any disk that is written to frequently is more prone to file fragmentation. By eliminating the fragmentation on the the Maverick BBS I was able to speed drive access by 30% (Estimate) and save a great deal of wear and tear on the drives. If you come across this problem, I'd recommend using DU-3 This utility will copy files, single or an entire disk to a fresh disk, one file at a time and greatly reduce fragmented files.

DU-3.BAS IS AVAILABLE ON THE MAVERICK

Well enough to chew on this month! Next month we'll tackle a few new commands.

****

NEW PJCCC ADDRESS!

Send all club correspondence to
145 7th Street, Morris Park
Phillipsburg, New Jersey 08865

We no longer use a Post Office Box!

And now a Great Thought From A Great Thinker
(suitable for framing)
(the thought, not the thinker)

In
Order
To Be Free
From
Flu
You Must
Flee
When
Flu Flies!

Bud Abbott

The above Great Thought By A Great Thinker
is presented as a public service of
The Penn-Jersey Color Computer Club
and
The 6809 Express
and
The 6809 Express Editorial Staff
all of whom extend to
You and Yours
a
Healthy and Happy
Holiday
Season
not to mention all the other
days of this or any other year
even
when it isn't a holiday!

FIRST CLASS MAIL

The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

6809 EXPRESS

The "6809 EXPRESS" is the official publication of the PENN-JERSEY COLOR COMPUTER CLUB. The club is based in the greater Lehigh Valley of Northeastern Pennsylvania including sections of Northwestern New Jersey. Any non-profit organization may reprint any part of this newsletter provided credit is given. PJCCC will gladly exchange newsletters with any other computer club. Send requests to EDITOR, 6809 EXPRESS, Penn-Jersey Color Computer Club, 145 Seventh Street, Morris Park, Phillipsburg, New Jersey 08865. PJCCC assumes no responsibility for errors or omissions. PJCCC assumes no liability for damages resulting from the use of any information or programs consined in this newsletter.

6809 EXPRESS

The Official Publication of The

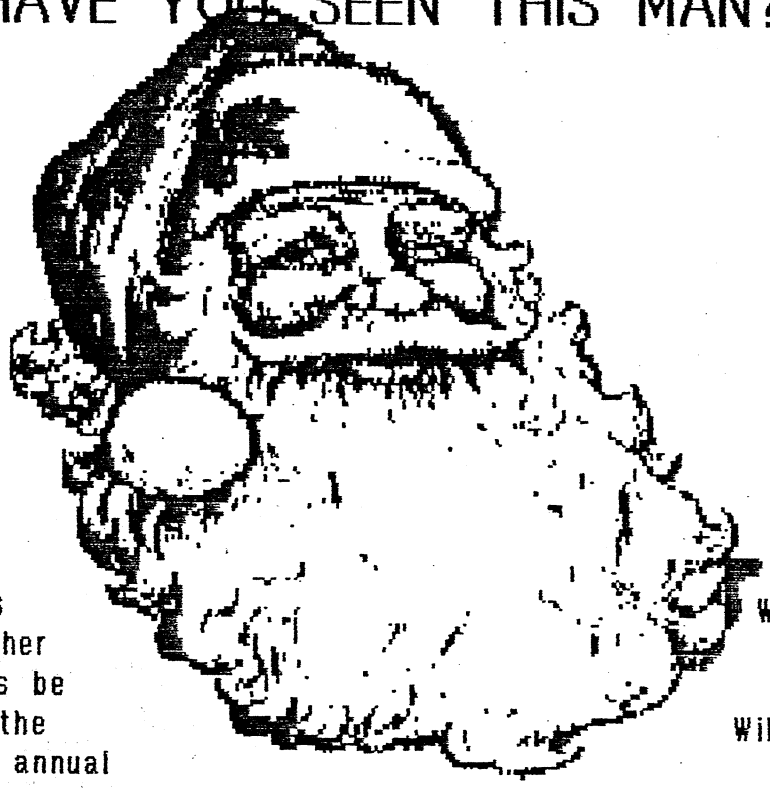
PENN-JERSEY COLOR COMPUTER CLUB

MAVERICK BBS

215-760-0456

DECEMBER 1991

HAVE YOU SEEN THIS MAN?



Did he bring Peter an MM/1 I/O board?

Do you like cookies?

Did he bring Peter a 105 meg hard drive?

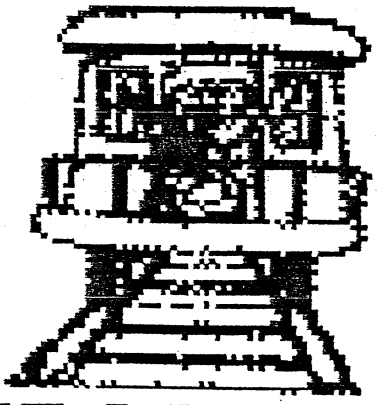
Do you like cake?

For the answers to these and other burning questions be sure to attend the exciting PJCCC annual

Will Clyde make the coffee?

Will you attend?

HOLIDAY PARTY FRIDAY DEC. 27 at 7 PM!



THE LIBRARY CAR

BY AL WAGNER

Oh Hi! Come on in, I've been working on a little gem for my word processing that I'd like to share with you. Pull up a crate over here by the computer and I'll continue my discussion on procedure files from way back in July.

If you recall, I mentioned that the startup file on a bootable disk is really a procedure file. This is a file that consists of a series of keyboard commands that execute as if you were typing them from the keyboard but, the computer is doing this automatically from the file. Well, two things have happened to me that have influenced my topic for this month. 1) I became more interested in what one can do with windows because of the last discussion we had and 2) I was writing some letters and was trying to write them in a formatted manner without having to run them through a formatter. To do this I needed a scale across the top of the screen to judge my margins. How could I create this scale without messing with my editor, have it so that it could not be accidently overwritten by the editor, and try to keep it simple? It dawned on me that what I needed was some combination of device and overlay windows.

Since this routine to create the window and scale would require several lines of commands to get the job done, I decided to create a procedure file to be sure that the commands were executed the same way each time. (I'm not the most accurate typist and my typing errors would drive anyone to the brink trying to debug a program if I had to type the entire program each time it was to execute!) I fired up my favorite editor and began.

I knew I needed to create a window. /W2 was available on my system, so I started with INIZ /W2. For no real reason, I decided to use wcreate to actually open the window. To create my favorite 80x24 white letter on blue background text screen I used Wcreate /W2 -s=2 00 00 80 24 00 01 01. Now we need to create the scale. I used two echo commands redirected to /W2. (Echo is similar to RSDOS BASIC's print command.) Echo >/W2 - 10 20 30 40 (etc. to 80) Echo >/W2 ----/----/----/----/(etc. to fill the line). With the scale created I now needed an overlay window in which to run my editor. This time I used display. Display 1b 22 00 00 03 50 15 00 01 >/W2. This opens an overlay window without saving whatever was under the area it is to occupy. It starts on the left edge of the screen, three lines down from the top. It is 80 columns wide (50 hex) and 22 rows long (15 hex) to allow for the two lines used by the scale. It will be a blue background with white letters. Next I needed to start a shell in this newly

created window environment. Shell i=/w2&. This is the way My file looks:

```
iniz /w2
wcreate /w2 -s=2 00 00 80 24 00 01 01
echo >/w2 -         10          20
   30        40          50          60
         70          80
echo                                  >/w2
----/----/----/----/----/----/----/----/----/----/----
/----/----/----/----
display 1b 22 00 00 03 50 15 00 01 >/w2
shell i=/w2&
```

Why not build yourself a file like mine, store it in your current data directory, and then call it by typing the file's name at the OS9 prompt just as you would a command. I'll wait. Hum dedum do doddle da de doe doo dum. Done yet? Ok, lets look at what you saw when you executed your file. First the disk drive came on, but you saw little else. After the last command executed, you saw a number with an ampersand appear below your OS9 prompt and then you got a new OS9 prompt. What happened? The new screen and shell were created and run in the background as far as your current screen was concerned. The number with the ampersand is the process number for the shell in our new window. By pressing clear you will get to the new window. How many times you will have to press clear depends on how many other windows you had open at the time you executed our procedure file and which one of those windows you were in. The new window is added to the end of the line. So if you had four windows open and were in the second window opened when you executed the file, you would have to press clear three times to get to the new window.
Go to the new window. Notice that the shell message is on the third line and that there is an OS9 prompt on the fifth line. You can now invoke your editor and you will have a scale across the top of the screen as a guide. Pretty neat, eh? We now have but one last problem. How do we clean up after we are done? We need to close the window. We could just execute the command for device window end, display 1b 24 >/w2. But this would leave a disconnected shell in memory. To dispose of the shell we must first be in the window with the shell we wish to terminate and then at an OS9 prompt type EX (cr). But now we can't type anything else in this window. Go to any other window and type display 1b 24 >/W2. This will then close the window.

Amazing as this may seem, I've actually been working on the Library this month! If you were at the last meeting you heard most of what I'M about to relate. I am undertaking to make disks of the collection of freeware, public domain, shareware, and home brewed programs that I have collected for my own use and add them to the clubs library. I currently have approximately 20 meg of this sort of program on my hard drive. Some of it is quite good and some, well, lets just say it leaves something to be desired. If anyone is looking for a specific type of program, let me know and I'll try to move that up on the priority list. I also have Access to software that is mine/yours for the for the getting. I have two lists, one for RSDOS and one for OS9 that combined have well over 250k of listing data. That's not the files themselves, that's just the listings that are that numerous. Each listing consists of one line giving the title of the file, the person who posted it to

the BBS where I can get it, and the date of the listing. I will be bringing these lists to the meetings for your perusing and will be happy to get any of the titles you would like to acquire. I must ask though that you follow the instructions that accompany each set of files as to shareware compliance and requests to keep groups of files together for any further distribution. Please bring a blank disk with you and I will return it with the requested items.

Oh no. I see Pete outside the Library waving at me through another window that he's got to put the express away for this month. Well, I won't be seeing you at the December meeting but I will be back in January. Try opening a few windows and let in some of that OS9 fresh air (its prewarmed by the power supply so you don't have to worry about your heating bills). Happy computing. Careful on your way out. One of these days I just gotta get this place organized

****

****

Presidents Report 11/29/91
Rick Hengeveld

I hope Santa was good to the PJCCC members this year! Since were looking 1992 square in the kisser, I guess it's time to start thinking about the coming years club activities.

Since there will be little if any old business to take care of at the Jan. meeting, we will be discussing and planning the upcoming presentations to be given at the 1992 meetings. So please give this some thought before the Jan. Meeting.

Secretary's Report 11/29/91
Rick Hengeveld (for Roni DeGarmo)

The November meeting of the PJCCC was called to order at 7:38 11/29/91 by the Club President. The treasurer's report was read and approved. No old business was pending. Under new business, Club librarian; **Al Wagner** reported that the club now has an up to date listing of all the files available on the **Delphi** network. The file is very extensive. Contact Al to receive the listing, also Al has offered to pull files from the system for club members. So make your requests known.

Al also stated he will be listing the club on the Delphi database. **Maverick** sysop **Rick Hengeveld** stated that he is close to having the necessary hardware to get a 65 Meg hard drive online to store the club files on our BBS.

Free sample copies of the 6809 Newsletter have been offered to potential new members. Also we may soon see a limited number of commercial ads in the 6809 to offset printing and mailing costs. *(next pg)*

(minutes continued)

The current slate of club officers were returned to office by acclamation. **Richard Kravits** stated he will continue to take care of our meeting area, by staying in touch with the collage. The meeting was adjourned at 7:57 and Random Access took place with a variety of subjects being discussed. A presentation was then given by Rick Hengeveld on writing programs in RS-DOS Basic.

The Maverick Report
Rick Hengeveld

The Maverick BBS has logged nearly 500 calls and has approximately 250 messages in the database. The files section has many new uploads there for the asking. The hardware is coming together for our hard drive system which will allow 2400 baud operation and mass storage of files. This should allow for greatly expanded use of the Maverick system.

****

****

And now the 6809 Express,always eager to expand the horizons of its reading public, presents another

GREAT THOUGHT BY A GREAT THINKER!
(this time in poetic form)

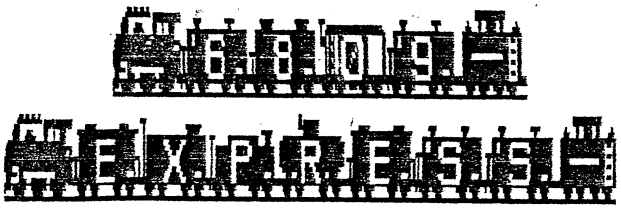
**I stood on the bridge at midnight
When I turned to myself and said
What a sap I was to stand there
When I could have been home in bed!**

Harry McNaughton
It Pays To Be Ignorant. circa 1945

****

****

Don't miss the annual HOLIDAY PARTY!
It will be a lot more fun with you there!
Bring some goodies to share if you can!
Friday December 27 at 7:00 PM



The Official Publication of The

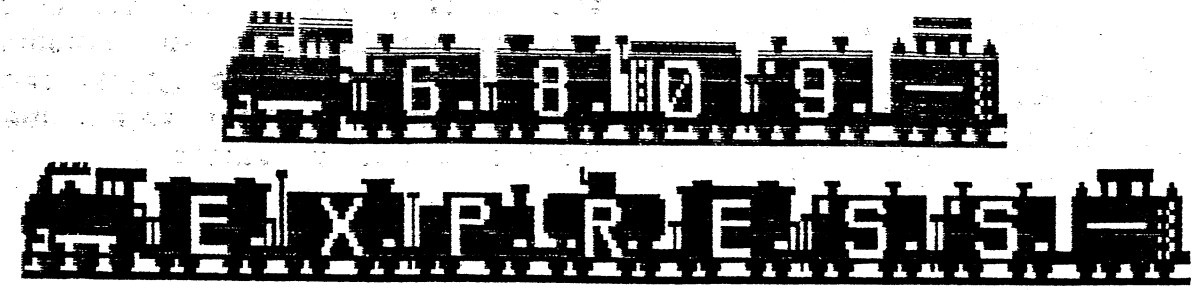
PENN-JERSEY COLOR COMPUTER CLUB

H. Peter Unks, Editor
145 Seventh Street, Morris Park
Phillipsburg, NJ 08865

Allentown PA 18103

FIRST CLASS MAIL

2912 1891 12/24/91



The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

MAVERICK BBS

215-760-0454

FEBRUARY 1992

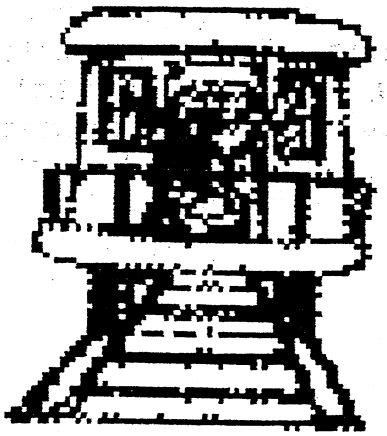
THE PRESIDENT'S REPORT
Rick Hengeveld

The countdown to the March PJCCC meeting continues. So what's so important about the March meeting? Well thats were we will pare down our club membership role to the active members. Currently we are carrying a number of people on our mailing list that are no longer active in the club. A lot of this is due to the fact that the PJCCC was able to forego club dues last year. At the same time some members moved away from the Coco, and are no longer active in the club. This year we've had to reinstate the dues ($15.00) and we will continue shipping the 6809 Express to only those who pay their dues by the March deadline. So if you can't make either the Feb. or March meetings, Let someone in the club know if you intend to remain in the club. Or better yet mail a check to the club. You can address it to the 6809 Editor, Pete Unks. Pete's return address is on the cover of your newsletter. Pete will see that your dues are proccessed.

****

SYSOP REPORT
Rick Hengeveld

The Maverick BBS has now gone over the 500 call mark! Over 100 downloaded files have been accessed from our database, All this from 40 registered users! For a Coco dedicated BBS, these are very good numbers. With almost 300 posted messages there has also been a great deal of information exchanged. For those few members that have a modem but have not yet accessed the Maverick, well ya don't know what your missing. Get online today!



THE LIBRARY CAR

By AL WAGNER

Welcome to the first session of the Library car for 1992. Hope you survived the holiday season in good shape. I'd like to talk a little on OS-9 I/O. OS-9 has a neat feature it inherited from its ancestor UNIX and that is standard input and output. What this means is that data coming into the computer and going out to a device looks the same to the computer regardless of the device from which it is coming or to which it is going. This simplifies writting many program functions. Let's say we wanted to move data from one place to another. The standard I/O feature means we need only one function that moves data from the standard input to the standard output to move data from the keyboard to the monitor or to move data from a disk file to the printer or from anywhere to anywhere. The default for the standard input is the keyboard. The default for the output is the monitor. Well, if these are the defaults, how do we use other devices? This is accomplished through the magic of redirection.

Let's say we want the date and time to show up on the printer. There is a date command in OS-9 that would display the date and time on the monitor. To redirect this to the printer the command would look like this: date t >/p. "Date t" is the command itself and as mentioned before this would put the information on the monitor. The ">" symbol redirects the output of a command to the device or file that follows it, in this case, the printer. Now let's imagine we've written a program where we enter data from the keyboard and it ends up on the monitor. We'll call this program KTOM. Now imagine we've created another program that generates the same data in a disk file called DATAFILE, that we had previously been entering from the keyboard. We'll call this program FILEGEN. KTOM also needs to be run several times a week. Typing all that data in accurately each time is difficult if not tedious. However we can redirect the input of KTOM from the keyboard to DATAFILE by the following command: KTOM <DATAFILE. The "<" symbol redirects the input of the program from the keyboard to the device or file that follows it.

Well that's just great, but now we want to feed KTOM directly from FILEGEN without having to use the disk in between. This calls for a pipeline from FILGEN to KTOM. In OS-9 a pipe is created with the

symbol "!". The command line would look like this: FILGEN ! KTOM. Redirection symbols "<>" are used between a program and a device. Pipes , "!", are used between programs. One caveat about pipes. In order to use them you must boot with Pipeman, Piper, and Pipe in the boot file.

Another use for pipes is in creating filters. A filter can be thought of as a program that modifies data. For instance a sort program. We've been looking at the output of our program KTOM, and have decided that it is getting difficult to find the entries we want for all the data. It would be easier to find them if the data were sorted. We have a sort program but don't want to take the time to manually run the output of the data from FILGEN through the sort but KTOM would work better if the data were sorted.

The easy way to handle this is use SORT as a filter between FILEGEN and KTOM. The command line would look like this: FILEGEN ! SORT ! KTOM. Now the output from FILEGEN is fed to SORT and in turn its output is fed to KTOM and we didn't have to reprogram anything! Isn't OS-9 NEAT!

Now look at another very practical place to use all of this. The command is DSAVE. Dsave allows you to copy large chunks of a disk, if not the whole thing, to another disk. The way it works is it starts at the current data directory and outputs commands to copy its contents and all directories below it to the specified location. The problem is it will output these commands to the monitor where they look impressive, but do nothing. You can handle this two ways, depending on what you need to do with these commands. Let's say you don't want to copy all the files but just most of them. We can redirect the output of dsave to a file that we can later execute as a procedure file. This way we can edit the file so only those files we want copied appear in the file. The command line could look like this: dsave /d0 /d1 >copfile. On the other hand if we wanted to let dsave copy everything it could, we could pipe the output into a shell like this: dsave /d0 /d1 ! shell. This cause each command to be executed by the new shell as dsave generates it.

Each command is displayed on the screen as it is executed, allowing you to follow its progress. But we are busy today! We don't want to wait for dsave to finish doing its task before we start another job. Another OS-9 symbol is the "&". Adding it on to the end of a command, such as either of the dsave command lines shown above, send it into a background mode and allows the screen to be free for other jobs. A typical command line might look like this: dsave /d0 /d1 ! shell&. If you recall, we used this last month to start our wordprocessing window without tying up the window from which we started our procedure file.

Another advantage to OS9 is the languages that are available. If you have a COCO3 and get OS9 level II, BASIC09 comes with it. Pascal, C, and Fourth are available. Since

BASIC09 comes with the level II package, let's talk about it for a while.

If you've done any programing in RSDOS, chances are you've programed in BASIC. The effort required to make step up to BASIC09 is quite small. You say you're not a BASIC programmer? Well BASIC09 fundamentals are no more difficult to learn than RSDOS BASIC fundamentals. The KEY operative here is "FUN"damentals. Besides once you get past the elementry beginnings of programming, you'll quickly appreciate BASIC09's quickness and power. The quickness over RSDOS comes from BASIC09 being a partialy precompiled language rather than a run time compile language as most other implementations of BASIC are. The power comes from its expanded command base. Variable structures are available that just don't exist in RSDOS. Direct access to OS9 commands and system routines is also there for the calling.

Let me explain about compiling. The microprocessor chip in any computer does not understand english or any other language, including BASIC! In order for the CPU to be able to process our instructions a translator, known as a compiler, must change our human readable instructions into machine instructions consisting of binary 1's and 0's. With most BASIC's, and RSDOS is one of them, this compiling takes place as the program is run. Computers can be very ignorant. Imagine yourself in a situation where you are in a foreign country and had to carry an English-Whatever/Whatever-English dictionery with you wherever you went. Imagine also that you couldn't remember a single word of this language even for a split second. Every time someone spoke to you, you had to look up every word. If that word occurred twice in the same sentence you would have to look it up twice! Now after a while you got very good at looking up words. So good in fact that most people didn't even realize you were doing it, but not as good as someone who did not have to look up the words. This is how a run time compiler works! Where the look up time really shows itself is in a loop where a program fragment must be executed many times in rapid succession. Each time through the compiler must re-compile each command in each line. By the way, the computer's act of looking up the commands is called parsing.

In contrast, a fully compiled language will go through the compile routines once at compile time and then when run will not have to look up anything as the translation to machine language has already taken place. This makes for very fast execution. BASIC09 is somewhere in between. Since a single BASIC command usually translates into several if not many machine language commands, a compiler actually looks up a routine rather than a single command. In a fully compiled language these routines are actually written into the compiled program. In BASIC09 however the compiling stops with an index to the machine language

routine being substituted for the command. This means the cpu jumps immediately to the ML routine without having to parse the command first. Its like having an ML program that consists of many subroutines. Where are these subroutines stored? Well if you run the module known as BASIC09, they are in there. If you went a step further and ran a routine known as PACK on the program, you could run the BASIC09 program from an OS9 command line as long as RUNB is present. RUNB is BASIC09's other repository for all the subroutines. You do not actually execute RUNB. You can preload RUNB and that will speed up execution of PACKed programs considerably. To execute a PACKed BASIC09 program, make sure it resides in the current execution directory and that RUNB is either also in that execution directory or loaded into memory, then type the name of the program as it appears in the directory. You don't need or want to type run or enclose the name in quotes.

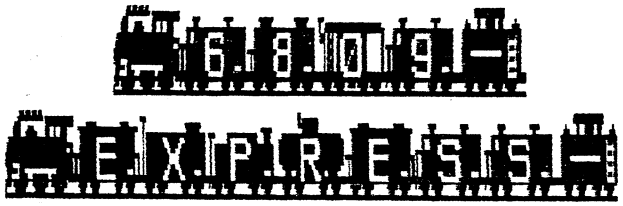
Next time we will try exercising a few simple programs. If you're interested, get a copy of The BASIC09 Tour Guide by Dale Puckett. It is a excellent tutorial on BASIC09.

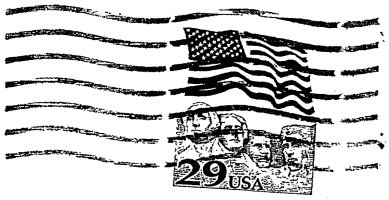
Well, that's enough to keep you busy for a little while. Cuddle up to that nice warm COCO power supply and try a few OS-9 commands. Anything worth learning takes a little effort. So far we are just scratching the surface. How would you like to be able to write machine language routines without having to learn machine language or even assembly language? With the OS-9 "C" compiler you can. The programs are even portable to the MM/1 or other 68k machines. They are even protable to mess-dos machines and mainframes! How's that for versatile? Well, you want almost machine language speed, but you've learned BASIC and you really don't want to have to learn another whole new language. BASIC09 will fill the bill. The COCO, particularly the COCO3, was made for OS-9. Come on, warm yourself with the joys of conquering new ground. Don't be intimidated by something new. I'll be seeing you at the meetings to answer questions and you can leave messages on THE MAVERICK or even call me at home. Try it. You'll like!

Happy computing!

****

FEBRUARY 28
FUN WITH
RASCAN PIX!!!
AND OTHERS TOO!
Dirty Trix with
CoCo Max III!
Peter Unks
will do the
DEMO!





The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

FIRST CLASS MAIL

6809 EXPRESS

The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

HAVERICK BBS

215-760-0456

MARCH 1992

TRENTON COMPUTER SHOW

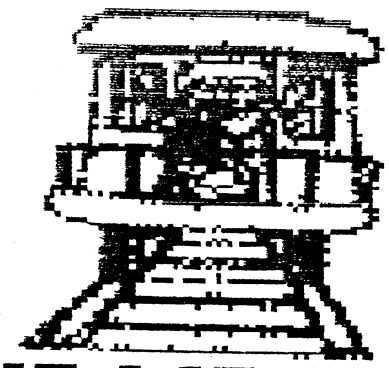
Rick Hengeveld

Well, spring is upon us. The weather is about to break, time for outdoor activities! Normally outdoor activities do not include computing. However April 11th and 12th have been set aside for outdoor computing! These are the dates for the famed **Trenton Computer Festival**, located at **Mercer County Community College**. While Trenton is far from a Coco related show, there are bargains galore for the Coco user! Almost all the items on sale are IBM related and that's not bad news for the Coco user. The big blue machines have a lot in common with the Coco. For instance modems, floppy drives, hard drives, drive cases and power supplies are all the same! Some monitors are also compatible. A good supply of blank disks will be available as well as RAM chips ect., ect. The list goes on and on. This year will be the fourth Trenton fair I've attended. I'm always impressed with the low prices and huge selection found there. The huge selection is courtesy of the 1000 table outdoor flea market. Both new and used equipment can be found in the outdoor market. Indoors you'll find 171 booths, selling anything from RAM Chips to complete state of the art systems.

Admission to The Trenton Computer Festival is $7.00 for both days or $5.00 for Sunday alone. Believe me the show is well worth the admission price.

****

PJCCC MEETS MARCH 27
RASCAN PIX AND COCOMAX III
CANON BJ-5 BUBBLEJET PRINTER



THE LIBRARY CAR

Alan J. Wagner

Welcome to the PJCCC Library car. Last time I promised a Basic09 program or two. So, let's get into it. Get your system booted up in OS9. Now, get your backup of the Basic09 disk or whatever disk you have your Basic09 module on. Insert it into one of your drives. Make sure you remember to CHD and CHX to the data and command directories (respectively) you wish to use on the disk you just put in the drive. Now type, BASIC09 (Enter). You don't have to capitalize the command as OS9 will find the module whether it is in capitals or not.

Also, just a reminder, (Enter) stands for the ENTER key. A similar symbol will be used for the spacebar. Your disk drive should have come to life and very shortly you should be looking at the Basic09 copyright message and a "B:". The "B:", minus the quotes, is the prompt for Basic09.

Just for fun press (Enter) and nothing else. What's this?? You got a directory of some sort. This is the module directory for Basic09. It is empty now unless you're ahead of me somehow. Notice it also tells you how much memory is available. Unlike OS9, we can ask for more room and actually get more! Let's try it. Type MEM (CR). Note that this time it only told us the amount of memory we have available. Now type MEM 10000 (CR). Note that now when it replied with the amount of memory it was close to 10k of memory available. The amount of memory you can ask for is dependent on how much you have in your machine and what else is going on in the machine at the time you ask for it. I have asked for and gotten 32k of memory with no problem in a 512k machine. You should remember that one can only get a maximum of 64k for any one process operating on a level II machine. Since Basic09 is itself a process, it and the processes under its control combined cannot exceed 64k. There is one way around this that we will cover later and I referred to it last time as PACKing. Another is to break your program down into modules that are called from the main program. Each of these modules are a process unto itself and can each use up to 64k! This too we shall see more of shortly.

Ok. We have now opened Basic09, seen that there are no processes currently in the Basic09 directory, and we have given ourselves a little more elbowroom with which to work. Let's begin entering a program. Type EDIT myprog (CR). Suddenly the familiar "B:" has become "E:". This is the editor's prompt. I am not going to go over all the editing commands but I will cover enough of them so that you will be able to get the job done. As always, once I've helped you get a taste for how to do things, I recommend that you read the manual for greater details. To enter a line of our program, press (SpaceBar) and then type the actual program instructions. The space indicates to the editor that you want to insert a line. To end a line, simply press (Enter). Back spacing will work

before you press (Enter). I'll cover how to edit a line after (Enter) is pressed and the meaning of the program commands after we get the program into the computer. Oh, two more things. Don't worry about capitalizing all the commands. Basic09 does that for you automatically! Also, DO NOT type in any line numbers. Basic09 doesn't need them and though Basic09 understands them, they just slow the program down.

Here is our first program.

```
DIM counter:INTEGER
DIM name:STRING[20]
PRINT CHR$($0C)
INPUT "Enter your name, please: ",name
PRINT
PRINT "Hello "; name; ". I am running Basic09!"
PRINT
PRINT "I can count to 10. Watch me!"
FOR counter:=1 TO 10
PRINT counter; " ";
NEXT counter
PRINT © PRINT
PRINT "That was very simple. I can also do more complex tasks."
PRINT "I hope you enjoyed your first program."
PRINT
PRINT "Bye for now!"
END
```

This program illustrates several of the features of Basic09. First let me give you a little more on the editor. If your fingers are as uncooperative as mine, you probably made a typo or two while typing in that program. If it was in a command itself, you got an error message as soon as you pressed (Enter). Not only did it tell you there was a syntax error in that line, but it pointed to where Basic09 felt the error was! How's that for convenient? OK. So it told us we can't type, now what? Well, let's say the error was that we also pressed the "E" when we went for the "R" in "PRINT". Something like this: PREINT. (This is one of the errors that seems to flow from my fingers with regularity.) As the first character of the line (no space as a space would tell the editor to insert the following as a line and we wish to issue a command) just type "c/RE/R (Enter)" and the "E" magically disappears. See Pete isn't the only one that can do magic! (Inside humor for PJCCC members.)

Let's go back and look at what we've typed, one line at a time. Type as a command, "-* (Enter)". This takes us back to the very beginning of our program. But wait! Where did those numbers at the left of each line come from? We didn't type those into the program! I thought you said Basic09 doesn't use line numbers! Calm down. The numbers are the offset address of the beginning of each line from the beginning of the program. The editor provides them as a reference. They are not actually part of the program. Once we leave the Basic09 editor, if you would "SAVE" the program and then call it into an OS9 text editor, those numbers would not be there. OK, now that we have that under control, press (Enter) and for each time you do, the editor advances one line in the program.

What's that? You say you accidently skipped a whole line when you were typing in the program? Index through the program until the line just BELOW the spot where the skipped line is supposed to be inserted is displayed. Now enter the line as you normally would. Now we want to back up a couple of lines to redisplay just the area where we expected our inserted line to be. Type "-3 (Enter)". This backs us up three lines. (Some of you may have noticed we used "-*" to backup all the way to the

beginning. You're right "*" is a wildcard! And for those of you who are thinking way ahead, yes, "+" will move you forward the same way as "-" moves you backward!) Now let's List these lines to see what we've done. Type "L5 (Enter)". This lists five lines. Ok, let's get drastic! Type "L* (Enter)". Wow, we listed the whole program. Hey, howcome that PRINT statement in the counting loop is indented more than the rest? When Basic09 lists a program "IF" statements and loop statements such as "FOR/NEXT" are indented to show which statements are included inside the conditional or loop controlled section of the program. What's that "*" on the left side of one of the offset numbers? My but we're full of questions! That's the pointer indicating at which line the editor is "looking". If you did a change command (c/mistake/correction), that is the line the editor would try to change. Inserts would go above that line. Etc,. You get the idea.

Now that we've gotten our program to look the way we want it, type "q (Enter)". If you didn't make any errors like leaving out the "NEXT" of a "FOR/NEXT" statement we should now be back to the "B:" prompt. Had you made such an error, you would have gotten an error message/s indicating the type of error by an error number and the offset from the beginning of the program where the compiler found the error. Yes, that's right, compiler. If you remember from last month, the compiler takes a first look at each line as you type (Enter) and the looks at the program as a whole when you exit the editor. If you got such messages, go back into the editor now and find the line in which the offset to the error occurs. Remember the numbers to the left of the line are the offset of the beginning of the line. If in fact it was a missing "NEXT", the error will be indicated as the "FOR" statement, as the compiler has no idea where you wanted the "NEXT" to occur. Make the corrections and then exit the editor. The rest of us will wait for you so take your time and try to get it right this time.

Type (Enter). Now our directory of procedures has a listing in it of the program we just entered. You should also see an "*" to the left of the listing. This indicates at which program Basic09 is "looking". If there were more than one program listed, the one with the "*" is the one that the editor would get if you just typed an "E". That would also be the one that will be saved if you type "SAVE (Enter)". Why don't you do that now to be sure that the program is saved for future reference. The program will be saved in the current data directory unless you specify a complete path list to another directory.

Now that we have our program saved, let's run it! Just type "RUN (Enter)". Basic09 will run the program indicated in its procedure directory with the "*". If there were any errors that for some reason we hadn't caught up 'til this time that caused the program not to be able to continue execution, the program halted with an error indication and you were introduced to the "D:" prompt. This is the debug mode and may be the subject of a later discussion. For the moment press (Break) and this will get you back to familiar territory and you can do your own debugging from there. I will assume error free execution and proceed with an explanation of the commands used in the program.

Basic09 REQUIRES ALL variables be declared at the beginning of the program and that the type be declared. Notice the first line. The variable "counter" is being declared as type integer. An integer is any number from -32768 to 32767. When creating loop counters integers are the way to go as they are incremented or decremented very fast.

In the second line, the variable "name" is declared as a string 20 characters long. Note that there was no "$" used in the string variable's name. You can use it, if you want for your benefit, but Basic09 does not require it In Basic09 at least the first 8 characters of a variable name are used by the program unlike RSDOS where only the first 2 are used. Capitalization also counts. If you have a variable "COUNT" and one "count", they are different as far as Basic09 is concerned, but I don't recommend it as it will confuse anyone trying to follow the program, including you!

Basic09 does not have a "CLS" command as in RSDOS. The third line is the equivalent. It is in fact a form feed, but it clears the screen and positions the cursor in the upper left. The rest of the "PRINT" and "INPUT" statements are the same as in RSDOS. Notice the line with two "PRINT" statements. In RSDOS, two commands can be on the same line separated by a ":". Basic09 requires a backslash. Though I know of no limit to the number of statements that can be strung together this way, try NOT to use this technique too often as it reduces the readability of the program if used to excess.

Well, I'm gonna call it a wrap for this month. I hope as always that I've kindled a little curiosity and encourage you to explore a bit deeper for yourself. My offer to answer questions via "The Maverick BBS" or phone still stands. I'll be seeing you at the meeting, so 'till then, Happy Computing!

****

DON'T MISS THE MARCH 27 PJCCC MEETING!

The Maverick Report

Rick Hengeveld

The Maverick continues to grow! Logging over 550 Calls and well over 100 downloads. Our message base has over 320 posts. There are about 40 programs available for download at this time. If your interested in downloading a program and you don't see it during your logon, then leave mail for myself or Al Wagner, our Club Librarian. Some members have had problems leaving private mail on the system. Private mail may be read by only the person sending the mail and by the specific person the mail has been addressed to. Nobody else has access to these private messages. To leave a private message the system requires the mail be addressed to the person exactly as his name appears in the userlog. Pressing "U" at the main menu will show you the complete userlog for the BBS. The only exception is when addressing mail to myself, either Rick Hengeveld or simply "sysop" will get the mail moved in my direction.

GET ONLINE WITH THE MAVERICK
BBS! DIAL 215-760-0456

6809 EXPRESS

The "6809 EXPRESS" is the official publication of the PENN-JERSEY COLOR COMPUTER CLUB. The club is based in the greater Lehigh Valley of Northeastern Pennsylvania including sections of Northwestern New Jersey. Any non-profit organization may reprint any part of this newsletter provided credit is given. PJCCC will gladly exchange newsletters with any other computer club. Send requests to EDITOR, 6809 EXPRESS, Penn-Jersey Color Computer club, 145 Seventh Street, Phillipsburg, NJ 08865. PJCCC assumes no responsibility for errors or omissions. PJCCC assumes no liability for damages resulting from the use of any information or programs contained in this newsletter.

The Official Publication of The
PENN-JERSEY COLOR COMPUTER CLUB

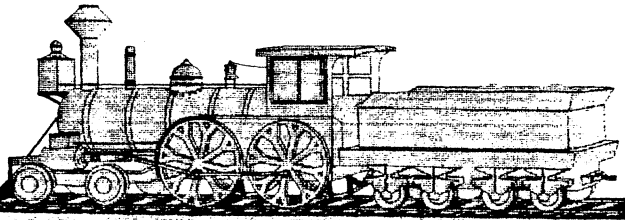
6809 EXPRESS

H. Peter Unks, Editor
145 Seventh Street, Morris Park
Phillipsburg, NJ 08865

FIRST CLASS MAIL

WAI T703 181032901 1891 03/25/92

JUNE 1994



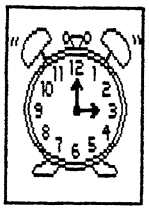
The 6809 EXPRESS

special all al Wagner issue!

NOTICE!
THIS WILL BE YOUR
LAST ISSUE IF
YOUR 1994 DUES
HAVE NOT BEEN PAID!

MINUTES OF APRIL 29,1994

Alan Wagner, Sr.



These are the minutes of the meeting of the PJCCC held on April 29, 1994. The meeting was called to order by **Rick Hengeveld** at 7:43 pm. **Clyde Gano** gave the treasurer's report and it was approved as read. There were no minutes of the March meeting as the secretary had not been in attendance and no one else took notes. There was no old business to discuss. **Richard Kravitz** announced that the meeting room has been secured for May, June, and July. A discussion of the program for the next meeting resulted in **Peter Unks** being on the hook with a demo of his newly acquired scanner. **Rick Hengeveld**, **Richard Kravitz** and **Clyde Gano** all announced that they would be unable to attend the May meeting. After some discussion, **Pete** said he couldn't see dragging the demo equipment to the meeting room just for himself and **Al Wagner**. He suggested that if we desperately wanted to have a meeting that I visit him at his house. It was decided that there would be no May meeting by a unanimous acclamation. **Rick Hengeveld** announced that the MSDOS version of the BBS has been on line now for one year. There have been about 1400 calls and there are about 100 users. **Steve Slagle** has uploaded over 250 Coco programs. **Rick** and **Pete** discussed the Coco Emulator for the MSDOS machines. **Pete** mentioned that he has a number of Appliance and Light controllers for the Coco. The meeting was closed at 8:12 pm.

BOOK REPORTS

Alan Wagner,Sr.

This report is a little different from that to which you have become accustomed in the Library car. Since I already had a program tutorial that had not yet been published, I thought I'd take the time to review a few books that I have read recently in preparing for the current tutorial being presented in the Library car. All but one of these are books I had mentioned at the beginning of the series on C. The second edition of **The C Programming Language** by Brian W. Kernigan and Dennis M. Ritchie published as part of the Prentice Hall Software Series is an update of the first edition to reflect the changes in the language brought on by the standards issued by ANSI (American National Standards Institute). Mr. Kernigan and Mr. Ritchie were responsible for coding the original version of C and their book's first edition was the "standard" for many years. In some circles, one can still here the original version of C referred to as K&R C. They quite humbly request that no one consider their second edition as the ANSI standard, but refer persons who would need to work with such a standard to ANSI for an up-to-date copy of the actual standard. The book is actually in the form of a tutorial in the language and much of what has appeared in this column on C so far, comes from that book. Reading and understanding the book from beginning to end would give one a very

complete view of the language. The viewpoint of the authors is unique because of their relationship to the language. Having invented the language, their perspective is one of knowing what was intended to be the language, as well as what it has become. Anyone seeking to learn C must read this book. From a historical viewpoint, it wouldn't be a bad idea to look up the first edition. It is out of print now, but if you are toying around with C and Microware's C Compiler for the Color Computer, this is the one you really need.

The second book is **C, a Reference Manual** by Samuel P. Harbison and Guy L. Steele, Jr. of Tartan Laboratories, published once again as part of the Prentice Hall Software Series. This book is truly a reference manual and the way in which it is constructed lends itself far more to looking up technical points on material with which one is already at least somewhat familiar, rather than a learning tool for the beginner. The material is extensively crossreferenced. At the end of the discussion of each of the various commands, functions, etc., there is a list of references to related topics. For example, some features of C almost never work alone but almost always require a companion command. Where such a feature is discussed the possible companion commands are crossreferenced. Another example is where a looping command is discussed. Similar looping commands are referenced at the end of each discussion. This manual covers both ANSI C and traditional C. It doesn't seem to be a good book for beginners, but it is an excellent reference once you've gotten a little experience under your belt.

Now we get to books that are truly for beginners. Peter Unks recently gave me a book called **C By Example**, written by Greg Perry and published by Que. I am truly greatful for his thoughtfulness. This book takes one from knowing nothing, to being able to handle some fairly complex operations. It gives a good step by step progression as it makes this journey from one end to the other. It is definitely aimed at someone working from an MSDOS platform and has quite a few references to the IBM style keyboards and commands. For those of us used to the longer filenames of UNIX and OS9, the 8 character cryptic MSDOS filenames take some getting used to. If you decided to use this book to learn C, you would most likely be successful. I say most likely as there are some significant editorial errors in the book. There was one spot where the author was making a point about one particular program feature and the program used to illustrate the point was incorrect on the very point he was making! There is another spot where the name of a file was obviously changed during editing, but was not changed in every reference to it. It was very confusing. The only complaint I have about the author is that he seems to completely skip command line arguments. Command line arguments are the ones that you give with a command, such as the MSDOS command, TYPE FILENAME. FILENAME is a command line argument. The examples he uses always start the program and then ask for the argument with a prompt. There is nothing wrong with this approach to programming, but teaching in this manner leaves one with the impression that C can't receive a command line argument when in fact it has a very rich way of handling this problem. In fairness to the author this may not have been his decision to leave this out. The list of editors at the beginning of the book reads like the New York phonebook. Perhaps it was a case of too many cooks in the kitchen each pointing at the other. If you purchase this book to learn C, you will get a good start in he language, but I think the next book is better.

This next book is **Teach Yourself C**, by Herbert Schildt, published by Osborne MacGraw-Hill. This book also takes one from beginner to well versed in the language. It is written in a step by step building sequence that teaches simple concepts in the early chapters and uses the knowledge gained to create a platform on which to learn more complex ideas. The example programs try to build a library of useful routines that you can draw on in future programming endeavors. The index is quite adequate and later becomes a useful tool in looking up commands and routines that you saw when reading the book. The authors style is clear and easy to understand. There isn't a whole lot more I can say about the book other than to say that it is a very good tutorial on C and deserves being read by anyone interested in learning C. That's it for now. Happy Computing.

THE TRENTON EXPERIENCE Of Alan Wagner

Well the time for the **Trenton Computer Festival** has come and gone. The following is a commentary on my experience this year. The alarm rang and started my day early so I would be sure to be on time to meet the gang for the trip. As I put the dog out for her morning constitutional, I noted that the ground was dry and it had not rained last night. It was comfortable outside and I was hopeful we would have a nice day in Trenton. After gathering my gear for the trip, I opened the door to begin to ferry the stuff to the car. No sooner had I stepped out of the house, it began to drizzle. I knew the time for Trenton was at hand.

As usual, the group met at the Seven-Eleven on route 309 in Quakertown. After deciding who was going in which car and getting coffee, we were off. Since I knew the shortest route to the affair, I lead off the caravan. One of the first-timers there stated upon our arrival, he'd have never found the place on his own. We reminded him that there are simpler but longer ways to get to the festival.

As we were directed to park in a lot that was very close to the back of the Mercer County Community College where the festival is held, Rick Hengeveld and myself elected to walk to the college from the lot rather than wait for the bus. We got about halfway across the athletic field when the heavens opened. We quickly ducked under a nearby shelter. The other part of our group that had elected to take the bus fared no better as they were still waiting for a bus when the rains came. When it let up a little, we continued our journey.

When we arrived at the ticket tent, the first bus had not yet made its appearance. Rick and I decided to wait in the tent for the others to arrive. By the time they got there, some five to ten minutes after Rick and I, the rain had let up a little more. We divided up into groups and set out in quest of our treasures. I can't speak for the others, but Rick, I and a young fellow Rick has been helping set out for the flea market section. After visiting the "outhouses" and a few booths, the rain began again in earnest. Our young fellow had not come prepared for the weather and had neither jacket nor rain gear. We headed for the shelter of the nearest booth with a covering. As we entered, the wind lifted the slack roof and dumped about a gallon of very cold water squarely down the back of our poor young man!

When the rain slacked off again, Rick and the young man set out to explore the inside displays. I returned to exploring the flea market. A short time later, I walked into a tent to see what was being offered there. Suddenly the wind gusted to the highest velocity it had been so far. The rain came down with a vengeance. Several people in the tent were holding firmly onto the supports as the wind was threatening to lift the tent and throw it into the next county. For a short while the tent supports were actually flying about an inch or so off the ground, held to this altitude by the people attached to the supports inside. Then as suddenly as it came the wind and rain went away. It actually became calm.

Later talking with Rick, he said they had been in an atrium area of the college and a sign painted on a 4x8 sheet of plywood had been launched into flight by the wind. Once the wind and rain had subsided, vendors could be seen putting the tents back together that had fared worse than the one I had been in. Some gave up on the tent idea as only shreds of the former structure remained.

The sun began to try to stick its face out and the festival began to settle down to its normal state of bustle. The "trunkload" vendors began to open their cars and the true flea market wares began to be displayed. Look as I would for Coco materials, I only found one box of joysticks and a couple of very old tape based pieces of software. I was able to find two 68b09ep chips at one table, but no Coco to put around them.

I purchased some tools and some other goodies and having depleted my allowance for this year, decided to drag my wet and hungry body back to the car for some eats. As I entered the parking lot, I immediately noticed that all was not well. There were four cars stuck up to their axles in the grassy lot now turned to a sea of mud! There was a tractor with a backhoe trying to coax one of them from the mire. I commented to a person nearby that this looked like it was going to be "fun" getting out of the lot. He said, "yeah." and pointed out that some of the more daring were still able

to make it out by traveling along the edges of the lot.

Not wanting to wait and see how bad it could get, I quickly got to my car and began to plan an escape route. All went well until I was almost out of the lot where some inattentive people walked into my path of travel and I had to stop. I almost gave up on ever seeing the car move again, but I determined to give it one more try. I backed the car up a foot or two and charged the little barrier that had formed when I stopped. The car gave a slight bump as it made its way over the lump. I was moving again!

With mud roostertails spewing from the rear of the car I headed for the nearest dry, hard pavement. I made it! Now I broke out the sandwiches I had brought for lunch. I didn't even want to get out to see how muddy the car was. As I drove and munched, I thought to myself how adventuresome the day had been. I hoped that Rick and the others would be able to get out of the lot as they were staying longer and would not be leaving for some time.

THE LIBRARY CAR

Alan Wagner, Sr.

This time in the Library car, we are going to modify the temperature program to make it a little more professional, faster and more accurate. First let's tackle an easy upgrade. The printout of the program isn't too nice looking as the numbers are left justified. The way most of us think with numbers is right justified. Printf() makes this easy to fix. Instead of printf("%d\t%d\d",fahr,celsius); we could printf("%3d %6d\n",fahr, celsius); This assigns a field width of 3 and 6 respectively. In addition, because we are specifying a field width, printf() automatically right justifies the field. We can negate this if desired simply by inserting a minus sign between the percent and the field width specifier. So now we've taken care of right justification.

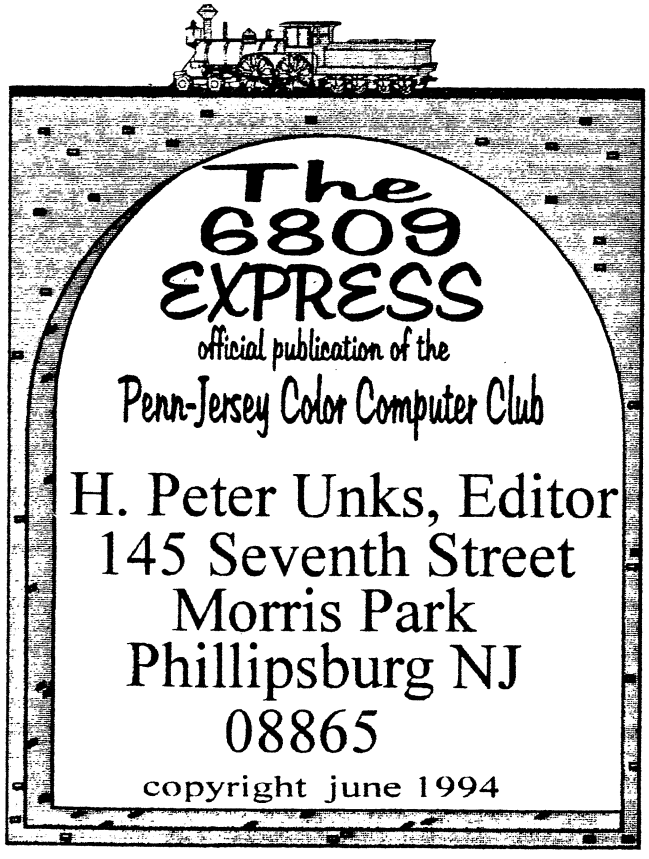
Next we want to address accuracy. If you've ever played around with converting fahrenheit to celsius, you've probably noticed that the numbers don't work out evenly as the program would imply. If we allow for one place to the right of the decimal point, that would probably be close enough for most of what any of us are doing. This however requires a significant change in the program. All the arithmetic was done using integers. Now we must use floating point numbers. A re-write of the program using floating point numbers might look like this:

```
#include <stdio.h>
/* print Fahrenheit-Ccelsius table
for fahr = 0, 20, ..., 300; floating-point version */  main() {
float fahr, celsius;
int lower, upper, step;
lower = 0;
```

```
/* lower limit of temperature table
*/ upper = 300;
/* upper limit
*/ step = 20;
/* step size
*/ fahr = lower;
while (fahr <= upper) {
celsius = (5.0/9.0) * (fahr-32.0);
printf("%3.0f %6.1f\n", fahr, celsius);
fahr = fahr + step;     } }
```

As you may notice, this is similar to the original program in many respects. Among the changes, fahr and celsius are now declared as being of type float. The next item that you may have noticed is that we were able to express the formula in a more "normal" way, ie., five ninths times the quantity Fahrenheit temperature minus thirty two. Notice that the five ninths is expressed as 5.0/9.0. The decimals and zeroes are required to let the compiler know that we desire floating point arithmetic in this case. Since without the added decimal points and zeroes, the constants are of integer values, the compiler would have assumed integer arithmetic, which would cause the result to be truncated to zero and all the answers to be zero in the table. Notice, if you will, the assignment "fahr = lower;" and the test "while (fahr <= upper)". In both of these cases, we have a floating point variable and an integer variable. In situations such as this, the integer is converted to a floating point number first and then the calculation is made. In the part of the formula that reads, "(fahr-32.0)", we could have left the decimal and the zero off the constant and the formula would have worked the same, ie., the constant would have been assumed as integer and then because fahr is a floating point variable, the constant would have been converted to floating point before the calculation. With the addition of the decimal and the zero, however, we avoid the overhead of the conversion and emphasize the floating point nature of the calculation to the human readers of the program.

The next change that was made is in the printf() statement. Notice that the %d's have been change to %f's. This tells printf() to expect floating point variables. In the first modification of the program, field widths were added to help straighten out the columns and make them more easily readable, but notice now a decimal and a digit were added to the field width specification. The digit to the right of the decimal point informs printf() how many digits to print to the right of the decimal when printing the floating point variable. In the case of the "%f3.0", printf() will print a field 3 digits wide with 0 digits to the right of the decimal. It will also not print a decimal point. In the case of the "%f6.1", printf() will print a field 6 digits wide and of those 6 the righthand most two will be a decimal point and a single digit representing a fractional part of the answer to the formula. It is important to note that the field width is for the entire field including any decimal or fractional part. Well that's it for this time. In the next installment, we will begin discussing the "for" statement and start getting into some preprocessor commands and how they can make a program more understandable and easier to change. Until next time, Happy Computing!

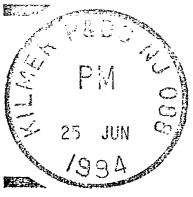


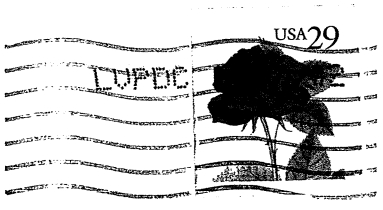
The 6809 EXPRESS

official publication of the

Penn-Jersey Color Computer Club

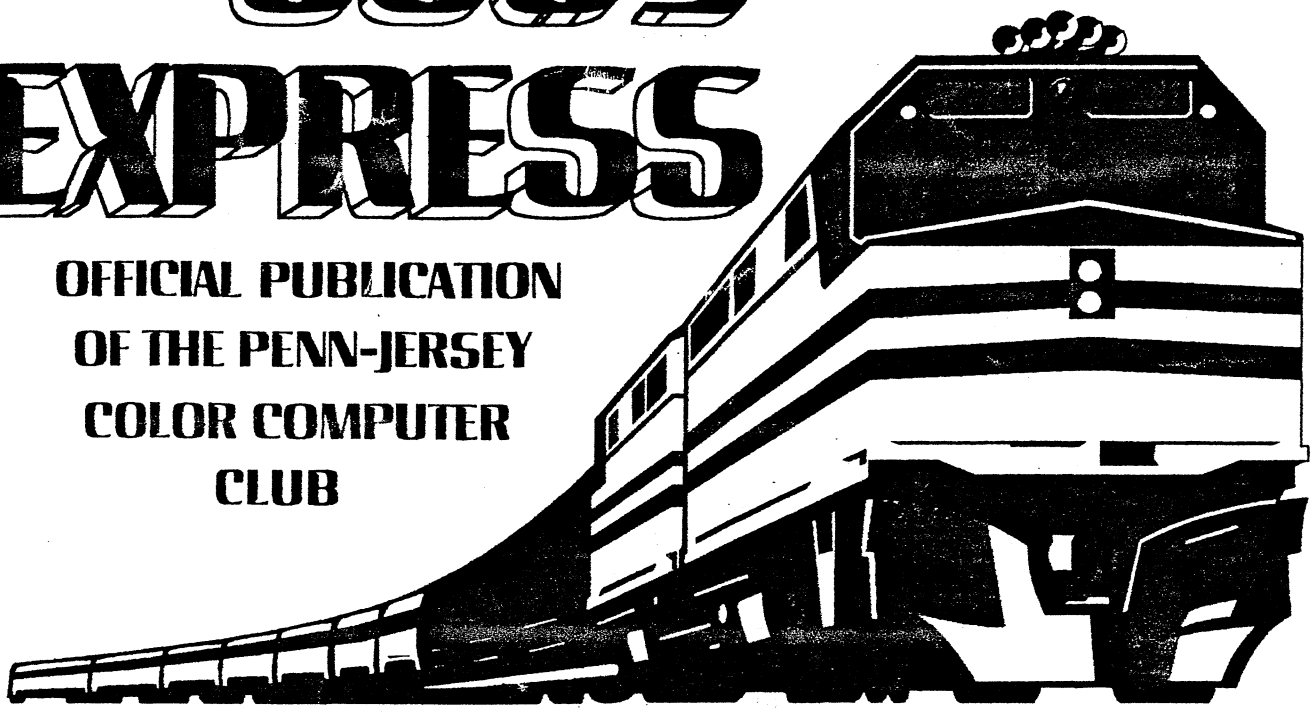
H. Peter Unks, Editor
145 Seventh Street
Morris Park
Phillipsburg NJ
08865





FIRST CLASS MAIL

The 6809 EXPRESS



OFFICIAL PUBLICATION
OF THE PENN-JERSEY
COLOR COMPUTER
CLUB

JULY / AUGUST 1994

IN THIS EXCITING ISSUE

UP-TO-DATE MINUTES OF JUNE AND JULY MEETINGS!

A REPORT FROM OUR PRESIDENT!

MORE ON C IN AL WAGNER'S LIBRARY CAR!

MINUTES OF JUNE AND JULY 1994

ALAN WAGNER,SR., SECRETARY

These are the minutes of the June 24, 1994 meeting of the PJCCC. Rick Hengeveld called the meeting to order at 8:00pm. The minutes were accepted as printed in the 6809 Express. Due to Clyde Gano's absence, Al Wagner read the treasurer's report. The report was accepted as read.

There was no old business.

New Business: Al Wagner read a letter of resignation for Clyde Gano. Due to his current state of health, he felt he could no longer carry out the duties of the office of Treasurer and with deep regrets, was resigning from that post. Rick opened nominations for treasurer. Eric Rhyder volunteered for the job. Pete motioned for the nominations to be closed. The motion was seconded. Rick called for a voice vote and Eric was elected by unanimous acclaim. Rick Hengeveld reported that the BBS has been on line now for over 450 days and has handled over 1600 calls. He is pleased with the performance of the BBS as the volume has been good for such a small, narrow focus BBS.

A discussion followed as to who would present the next demonstration. Peter Unks volunteered to show off some of his T&D software collection.

Random access was called. As usual a lively discussion of a widely varying range of subjects occurred. The meeting was closed at 8:30pm so that Pete could begin his demonstration of his scanning software and hardware. It was quite interesting and an enjoyable time was had by all.

These are the minutes for the meeting of the PJCCC held July 29, 1994. The meeting was called to order Rick Hengeveld at 7:30pm. The minutes of the June meeting were read by Al Wagner and accepted as read. The treasurer's report was given by Eric Rhyder and accepted as given.

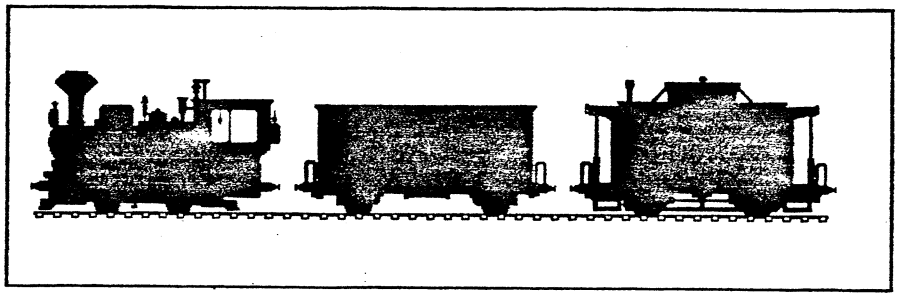
Once again, there was no old business.

New Business:
Richard Kravitz reported that he has taken care of the meeting room arrangements through December. He also, once again, gave a donation in the name of the club to secure the room. Rick Hengeveld gave the sysop's report on the status of the BBS. The BBS has now exceeded 1750 calls. He mentioned that he is open for suggestions as to how we might improve the BBS or any changes that may be needed. He also reported that the BBS was down for a short while during the month due to a power outage at his house. It seems if the power is off for several seconds, the system is able to recover by itself quite nicely. However, if the power is off for only a fraction of a second, the power supply in the computer folds back and will not come up without turning the main switch

Clyde has always gone above and beyond the call of duty for the membership. He has also donated a lot of time and equipment to getting others more unfortunate the himself started in computing. Truly the PJCCC has become a better organization through his work and input into both our club and each of our lives. We all hope to continue to see Clyde at as many upcoming meetings as he can make. There will always be a chair for Clyde Gano at any PJCCC meeting.

THE LIBRARY CAR

ALAN WAGNER, SR.



Welcome once again to the Library car of the 6809 Express. As promised, this time we will be discussing the "for" statement and some preprocessing commands. The "for" statement is similar in operation to the "for" in BASIC. One sets up a variable to be a counter, checks to see if the variable has reached some preset value before executing the loop, and then changes the variable each time through the loop. The changes in the variable can be by incrementing, decrementing, or modifying the variable in some formula. BASIC usually uses one of the first two. The physical way in which the loop is constructed is very different All three of the "for" statements function occur on one line in C. Assuming "x" to have been previously declared, the following could be a "for" statement used as a time delay. It would do nothing but loop around on itself very fast.

```
for(x=0;x<=1000;++x);
```

Notice how the statement is constructed. The line begins with the reserved word "for" to announce what operation is to be done. Next we find a parenthetical section. This section is divided into three portions separated by semicolons. First we find "x=0". This is where the initial value of our variable "x" is set, in this case, to zero. The variable could have been set to any number within the range of the variable type chosen for the counter. Next we encounter the checking portion of the statement. Here "x" is compared to 1000. This could be any kind of comparison that results in a true or false outcome to trigger the exit from the loop. As long as the outcome is true, the loop executes again. The next section is where the statement increments "x". Again, this could decrement "x" by putting a -- in place of the ++ or this could be some complex formula that calculates the new value of "x". This "for" statement is a null loop. That is, it doesn't have any statements that it executes outside of itself. To allow "for" to do bigger and better things the construct can be modified as follows:

off, waiting a few moments and then re-establishing power. The result is if such a momentary power outage goes unnoticed in his household, the system is down until someone happens to notice its inactivity. We discussed Uninteruptable Power Supplies, but the problem is the expense involved. A soon to occur change in the physical layout of the Hengeveld's household layout should bring the BBS system together with Rick's other computers and this should make for quicker recognition of BBS problems.

At this time Rick called for a motion to close the meeting. Al Wagner made a motion to close the meeting and Steve Slagle seconded the motion. The meeting was closed at 7:45pm.

The random access that followed was interesting and varied as always. After the discussion, Peter Unks gave a very interesting demonstration of two pieces of software for the Coco 3. The first was a replacement disk based software from Hawksoft for the old Plug-n-Power controller cartridge. The cartridge never worked on the Coco 3 due to the type of graphics used. Pete was as entertaining as always. The second software was an OS9 database that is now apparently orphanware. It was a very interesting demonstration.

OUR LEADER SPEAKS!

RICK HENGEVELD, PRES.

The June PJCCC meeting was quite interesting. H. Peter Unks demonstrated his color sheet feed scanner, quite an interesting gadget! Feed in a sheet of text, a photo, whatever you like and bingo it's digitized and up on your screen. From there you can do just about anything with the image. The possibilities are endless!

While the meeting had it's usual amount of fun and frivolity, there was a sad note. I must reluctantly announce Clyde Gano's resignation as Treasurer of the club. Clyde has been in ill health for some time and now he's requested a rest from his treasurer's duties. Eric Rhyder has now assumed those duties. It seems Clyde has been around since the very beginnings of the club, of course I've accused him of being around at the beginning if time! (Big Smile!)

```
for(x=0;x<=1000;++x){multiple; statements; for the loop;  to execute; }
```

Notice the braces. These allow for multiple statements to be included in the "for" loop. Everything from the opening brace to the closing brace is part of the loop. This is similar to the braces that open and close the "main" function. Braces are used many places in C programs to mark the boundaries of a block of statements. In most cases, the entire block can be viewed as one complex statement. The individual statements within the block still execute as individual statements, but to statements outside the block that call that block, it looks like one statement. Keep this in mind as we will be seeing this again in several discussions in the future. Those of you who may have some knowledge of C may call my attention to the fact that the "for" statement can have one other statement besides itself in a loop simply by leaving off the semicolon after the "for's" parentheses. This is true and in keeping with what was said in the above paragraph about blocks. The braces and everything between them is viewed by the "for" statement as the one statement that is allowed after the "for" statement itself. It is not a bad idea to get in the habit of including the braces around the statements in a loop, even if technically they are not needed. They are permitted in every case and are far less likely to be forgotten in the cases where they are needed if they are always used as a matter of habit. Two more important points before we continue. In the variable changing portion of the "for" statement, the increment, decrement, or whatever formula is used, the operation is ALWAYS done AFTER the loop has been executed and BEFORE the comparison to determine if another pass through the loop is needed. Therefore, it does NOT matter if ++x or x++ is used. The increment will take place after the execution of the loop and before the comparison. The second point is that any or all of the portions of the parenthetical statement can be omitted. For example: "for(;;);" is a valid loop! Since there is no counter variable initialized, no comparison, and no changes to the nonexistent variable, this loop goes on forever! Usually such statements have a block of statements included in the loop and somewhere in that block there is a conditional statement containing a "break;" or an "exit;" statement to exit the loop. In addition to an infinite loop, leaving out just the initialization portion of the "for" statement allows one to use as a counter a variable set to some number by calculations, keyboard entry, or such before the "for" statement was encountered. We will cover this in more detail as the circumstances arise. Let's take a look at one possible way our program could appear using a "for" loop.

```
#include <stdio.h>
/*print Fahrenheit-Celsius table*/
main()
{
int fahr;
for (fahr = 0; fahr <= 300; fahr = fahr + 20)
printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));
}
```

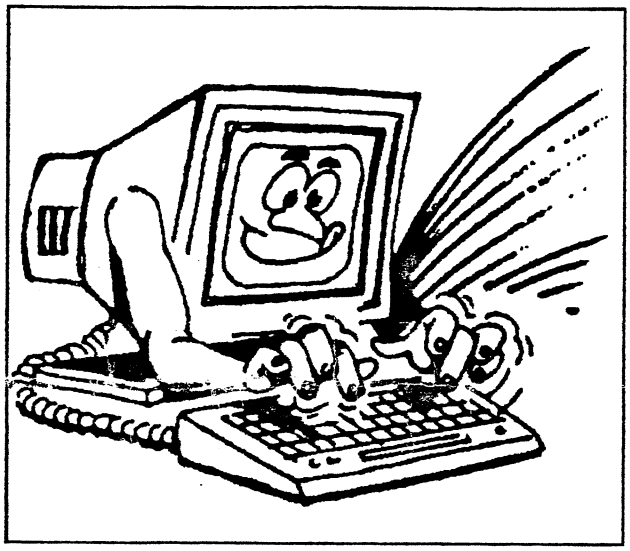
Wow! Did that shorten things up! Let's dissect this new listing a bit. First, we've eliminated the variables celsius, upper, lower, and step. Only fahr remains and this we have returned to an integer. Upper, lower, and step now appear in the "for" statement as constants. (Maybe not a good idea. More on this shortly.) Where the variable celsius had been used to hold the results of the conversion calculation, the formula itself now appears in the "printf" statement. This illustrates a general rule.

Any where one could use the value of a variable, one can use an expression of the same type. An expression here meaning a formula or calculation. Even functions returning an appropriate type could be used in such a context.

About two paragraphs back, I said that it might not be a good idea to have used constants in the "for" statement. We have already seen how a variable could be used in this situation, but variables have a way of becoming hard to find in a program. Tracking where they may be changed may be even harder. If what we need is really a constant that we can find and change easily, a symbolic constant may be in order. We have already used one preprocessor command, that is, "#include". This allowed us to include the standard I/O header file as if it were part of our program without our having to type it physically into our file. The next command for the preprocessor we will look at is "#define". "#define" allows us to create symbolic constants. Rewriting our program using this technique could make it look like this.

```
#include <stdio.h>
#define LOWER 0
/* lower limit of table */
#define UPPER 300
/* upper limit */
#define STEP  20
/* step size */
/* print Fahrenheit-Celsius table */
main()
{
int fahr;
for (fahr = LOWER; fahr <= UPPER; fahr =
fahr + STEP)
printf("%3d %6.1\n", fahr, (5.0/9.0)*(fahr-32));

{
```



LOWER, UPPER, and STEP are not variables. They are constants. Each place in the program where one of them appears, the appropriate value is inserted in place of the symbol when the preprocessor prepares the program for the compiler. Because they are symbolic constants, it would be an error to attempt to declare them anywhere but in the "#define" command lines. Two advantages for using symbolic constants are 1) appearing at the beginning of the program, they are easy to find if changes are needed and 2) a numerical constant buried in a program is not always clear about why it is there. UPPER is far more clear as to its function than the number 300. The same can be said for LOWER and STEP.

This concludes our discussion of temperatures for now. If there are any questions about programming in C (BASIC, Pascal, or etc.) leave me a message on the club's BBS. I'm usually up there on weekend mornings. I will answer you either on the BBS, if you need your answer quickly, or in the Library Car if the question is such that the group would benefit from the answer and you are not in a big hurry for the answer. Until next time. Happy computing!

Call THE MAVERICK BBS at 1-610-760-0456

The 6809 EXPRESS is the official publication of The Penn-Jersey Color Computer Club. The contributions remain the property of those who authored them.

The Penn-Jersey Color Computer Club is a not for profit organization.

FOR THRILLS! CHILLS! AND MILE-MINUTE ACTION- DON'T MISS THE NEXT MEETING ON SEPTEMBER 30, 1994 AT 7:00 PM!

A SPECIAL SUPRISE GUEST WILL BE WITH US!

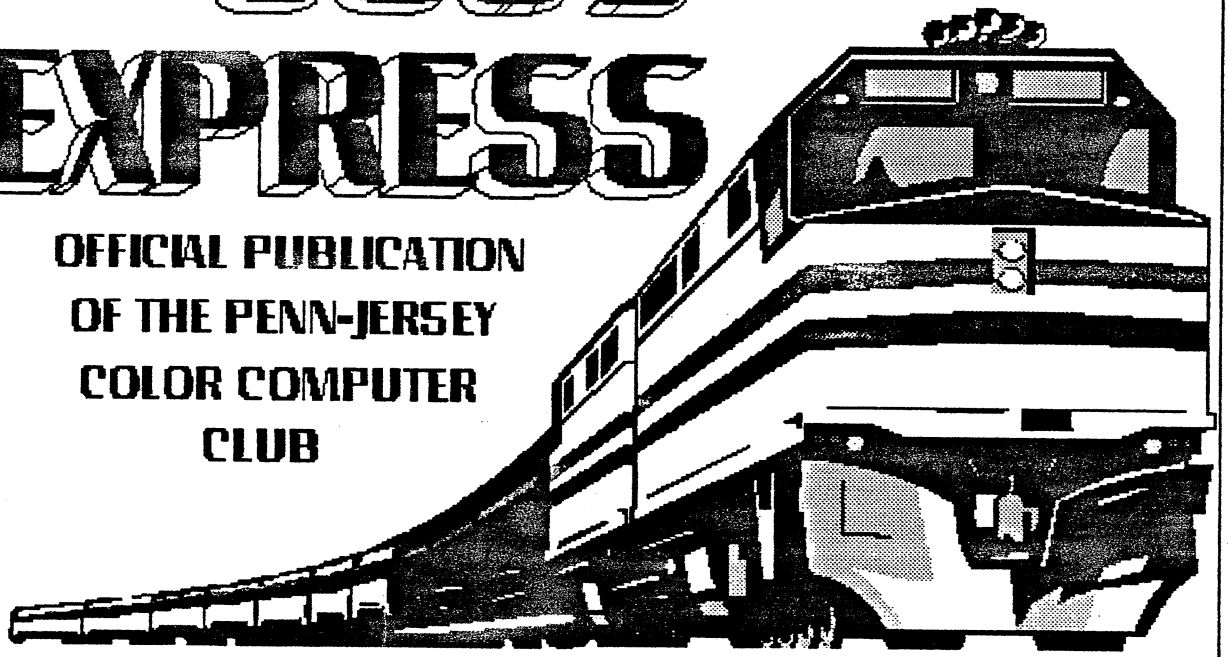
Clyde Gano had the good fortune to meet **Ruth Longacre** when they both worked for Hauser Chevrolet. Clyde was the parts and service manager. Ruth was secretary and keeper of the books.

Better yet, he had the good sense to ask her to marry him. She agreed. The wedding took place on March 25, 1944.

So now we know the reason for Clyde's big smile for which he is known far and wide.

HAVE LOTS OF FUN! PLAY THE CLYDE AND RUTH GAME! HERE'S HOW TO PLAY! WRITE THE WORDS "CLYDE AND RUTH" AT THE TOP OF A SHEET OF PAPER. SEE HOW MANY WORDS OF FOUR LETTERS OR MORE YOU CAN MAKE OF THE LETTERS IN THOSE WORDS. THERE ARE 842!

The 6809 EXPRESS

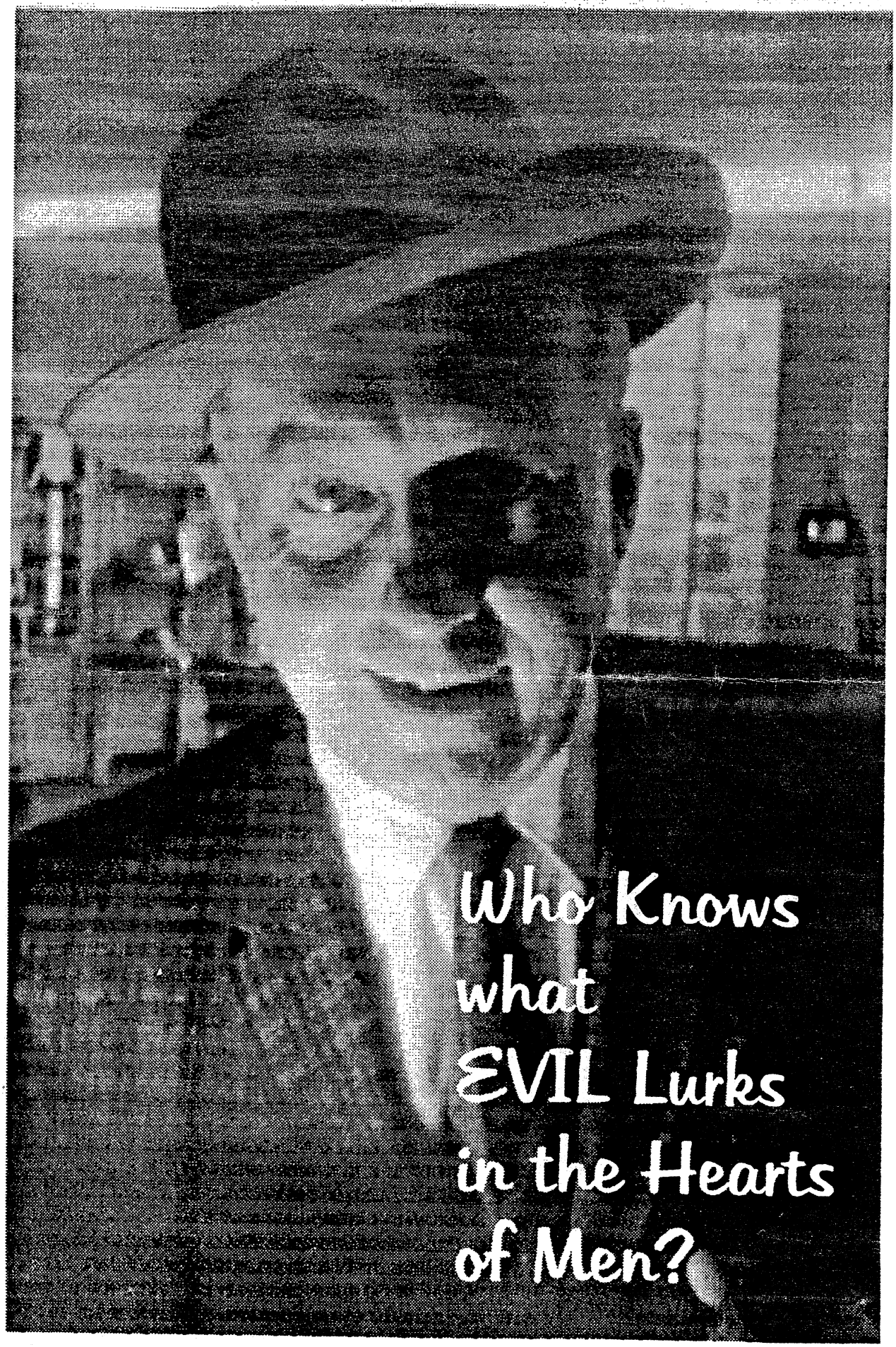


OFFICIAL PUBLICATION
OF THE PENN-JERSEY
COLOR COMPUTER
CLUB

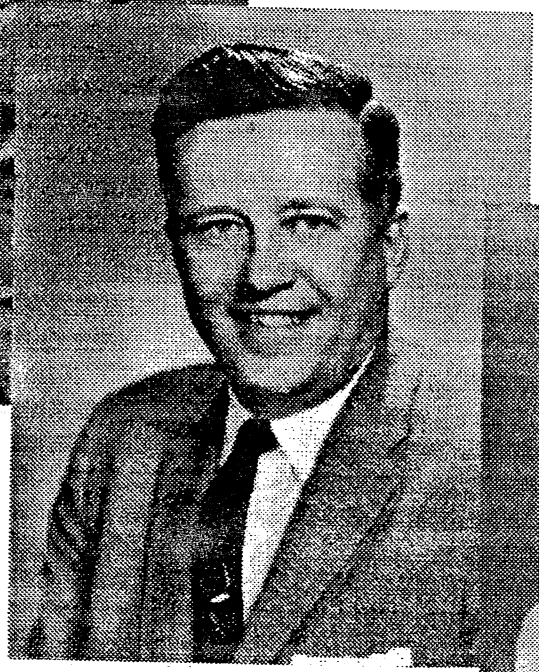
CLYDE GANO 1994

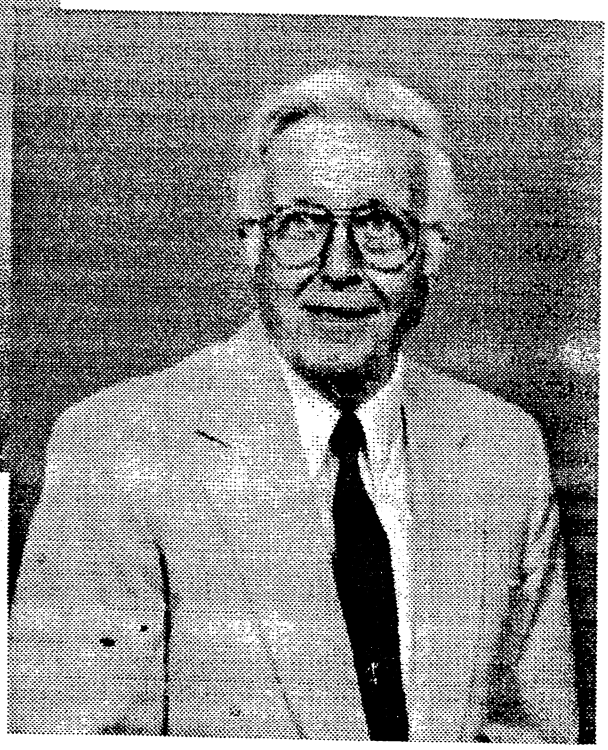
IN THIS ISSUE;

CLYDE, Who Else?
O.K., Ruth, but
that's all except
maybe minutes!


Who Knows
what
EVIL Lurks
in the Hearts
of Men?

The EVOLUTION of CLYDE GANO





Call THE MAVERICK BBS at 1-610-760-0456

The 6809 EXPRESS is the official publication of The Penn-Jersey Color Computer Club. The contributions remain the property of those who authored them.

The Penn-Jersey Color Computer Club is a not for profit organization.

The 6809 EXPRESS

H. Peter Unks, Editor
145 Seventh Street, Morris Park
Phillipsburg, NJ 08865

FIRST CLASS MAIL

These are the minutes of the August 26, 1994 meeting of the PJCCC. (ALAN WAGNER,SR.)

Due to the unexpected absence of Rick Hengeveld, the meeting was a bit less formal than its routinely casual execution. As Rick was still among the missing at 8:00pm, Clyde Gano gave a report on a list of Coco III users that was compiled by a disk magazine for Coco users. It listed between 200 and 300 Coco enthusiasts, with some in our area that Clyde has attempted to contact and encourage to join our happy group.

Peter Unks suggested that we should have a somewhat more formal meeting. He dutifully called that meeting to order at 8:05pm.

Pete called for an inspection of the minutes of the June and July meetings. Clyde made a motion to approve the minutes as printed and Richard Kravits seconded that motion. The motion passed unanimously.

Eric Rhyder gave the treasurer's report. Pete called for a motion to accept the report. Al Wagner motioned it be accepted as read. Richard Kravits seconded the motion and the motion passed unanimously.

Pete next called for any old or new business. There was none that anyone in attendance could recall. Pete then called for a motion to close the meeting. Clyde Gano motioned to close the meeting. Al Wagner seconded the motion. The motion passed unanimously and the meeting was pronounced closed at 8:12pm.

As Rick Hengeveld was also to have given the demonstration, we continued in the random access mode we had been in before Pete called the meeting to order. As always, a wide variety of subjects were discussed. Near the end of the discussion, Pete announced that there
would be a secret guest of honor next month and that none should miss the occasion. We all wondered who this could be.

President's Report Sept. 1994 (RICK HENGEVELD)

Hello all! First my apologies for missing the last meeting, as you know with a job of my type schedules are sometimes impossible to keep. While I'm on the subject of schedules, I won't be able to attend this months (Sept.) meeting. I will be off on a much needed vacation.

It wasn't very long ago that people who had a PC in their home were considered to be slightly eccentric, OK we oddballs! After all what could we hope to do with all that printed circuit junk! Computers, we were told were the domain of the business user and weren't meant for the home user as anything other than a curiosity. Well I'm here to tell ya that the home PC has finally landed as a firm piece of Americana. So what qualifies that contention? Do I sight some Govt. study? Maybe some private sector think tank? Maybe a United Nations proclamation? Naww, I simply turned to that great American arbitrator of what's in and what's out, the Television! During my travels I'm required to stay in a motel twice a week, and while there I'm treated to The Jones Computer Network. JCN as they call themselves is a 24 cable channel dedicated to nothing but computer technology. The best description I can give is to say it's The Computer Chronicles run amuck! So is this JCN a bonafide cable channel? Well I put it to the acid test and sure enough at 3:00 am. up came the infomercial for engine oil additive followed by another hawking some kinda fantastic new kitchen mop. Yeah by today's standards that qualifies JCN as one of the many new cable TV options. Seriously JCN will provide you with a ton of information and it's also a good place to get a look at many new products since they are almost continually demonstrating soft and hardware. I don't know of an area cable company that is carrying JCN yet, but it may only be a matter of time so keep your eyes peeled.

See ya next month!

Sysop Report Sept 1994 (RICK HENGEVELD)

To date the Maverick BBS has logged over 1900 calls to the system and supports 115 users. Over 300 files are available on the system. Currently the system is running on a 40 meg HD. After about a year and a half service storage space is starting to become a problem. We are currently running below 10% empty space on the drive system. While I haven't requested users to upload files, many good people have been charitable and provided a number of files to the system. So if your in need of files either for your DOS or your Coco stop by and have a look. Also if you currently use a DOS machine have a look in the MSDOS utility section. Check out the Coco emulation program for MS-DOS systems. The current version emulates Coco 2 and runs OS9 Lvl.1 A 386 DX40 will run in Coco 2 mode at about the same speed as the original Coco hardware, and a 486 machine really makes the old Coco2 software scream!