6809 EXPRESS

The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

MAVERICK 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.

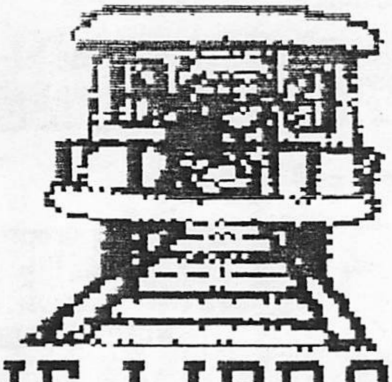
Admission to The Trenton Computer Festival is $7.00 for both days or $5.00 for Sunday alone. Believe me the show is well worth the admission price.

****

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 RSDOS, 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

[redacted]

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.

6809 EXPRESS

The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

6809 EXPRESS

The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

H. Peter Unks, Editor

[redacted]

[redacted]