

NEWJUG 99ER'S NEWS

SEPTEMBER 1992



SEND TO ONLY ONE

Highlights:

MOOS Conference	02	Comments & Suggestions	20
VAPP to Page Pro	13		
Hacking in c99	15		
Editor's Forum	16		
Joke of the Month	17		
99/4a & 9640 Vendors	18		

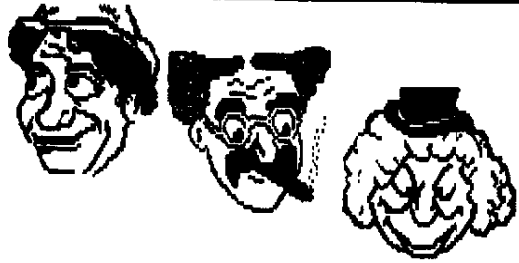
The opinions expressed in this newsletter are not necessarily those of the officers or members. Articles printed in this newsletter may be copied or reproduced with the proper credit to the author and the newsletter.

NEWJUG 99ER'S UG
P.O. BOX 1463
SAVREVILLE, N.J.
08871-1463

FROM:
NEWJUG 99ER'S UG
P.O. BOX 1463
SAVREVILLE, N.J.
08871-1463

POSTAGE GOES HERE

TO:
AFFIX MAILING LABEL HERE



MDOS
DEVELOPMENT
CONFERENCE
#3

July 14, 1992

Attendees:

9640NEWS	Beery
JHWHITE	Jeff
EICHER	Dan
CAL47	
DONEIL	Don O'Neil
BBDOONE	Barry B
CLINTP	Clint
BARNESW	Warren
ABEARD	Al
JERRYVC	
GREGPH	

<Dan> Hi Clint.

<Clint> Hi all!

<Dan> hey Clint, ever you ever looked at the source code to the MicroEmacs editor?

<Clint> Yes, I've ported it to two unix systems at work - but I prefer QDE!!

<Dan> Hum, could that be why QDE is available for the 9640 and emacs isn't? If you haven't guessed, I am biased towards emacs, I run it in dos and SCO xenix.

<Clint> Also, u-emacs is much bigger and would be slow on a 9640.

<Dan> even if a person was running in single user mode? :)

<Clint> I used to use it on an XT and the cursor was always running "late".

<Jeff> Al said he might not make it, so I guess we are ready to start. Where did we leave off last time?

<Beery> I think we left off last time setting up a "bug report" list.

<Jeff> How is that going?

<Beery> Jerry did a good job setting it up, but i've only gotten 2 or 3 responses, and those were really the "wish list" stuff.

[Editor's note: - credit for the bug report setup belongs to Jeff Guide and the Delphi programmers who set up our message-generating system]

<Dan> Over a year ago, I reported some bugs to PaulC, he said he had found and fixed them after a couple of message exchanges... So... I guess iffen you get a ll versions of the source, your will get those bug fix's.

<Beery> Well to date, I don't know what those "fixes" were. If you still have that list, "checking" it out will be helpful.

<Clint> I have two well-documented problems (heated exchanges with PaulC) on file here. Both are i/o related.

<Jeff> Perhaps we should catch up on how the effort to get the source code is going. Beery?

<Beery> Clint, send me (here/bbs) those bugs and I will put it all together when the source arrives. Source status.... When I last spoke with Lou (2 to 3 weeks ago), he was going to try and get something to me to "whet" my appetite, but it was dependent upon Paul returning from an out of town trip and Lou was fixing to go travel to S.America to adopt another child. He was expecting to be out of town a couple of weeks. I tried calling him today, but got no response. When we last spoke though, Lou indicated he was acting as a "go-between" between me and Paul and Paul was satisfied with a \$3000 tag.

<Beery> To date, I am fairly sure (haven't run a total in the past month) that I've raised \$3000, but not sure of travel expenses, might be a little short. That's where I am saying about "5%" short so the last few donations that haven't come in will complete it along with some of the small expenses (phone , etc) that I have incurred to date. If Lou is back from S. America, and I am not sure he is yet, I would expect a couple of weeks "recovery" on Lou's behalf before I can travel north to get the code. That's where she be. ga

<Beery> back in a sec, gotta get my dinner out of the oven. It will take 2 minutes.

<Jeff> Beery, perhaps you can remind us what parts of MDOS each of your team members will concentrate on.

<Beery> Hmm, good question.

<Jeff> Team members being Al, Barry, Clint, and you.

<Beery> Clint is going to be working on Vid xop's Barry B has expressed strong interest in DSR area Al Beard with Memory management and Math

<Barry.B.> <--- DSR's and I/O of all kinds :D

<Beery> Myself with the CLI and some of the vid/cli/memory code to support multi-tasking properly.

<Beery> Jeff White for taking the 1400 sector source file and transforming it into a 6000 sector source file with assembly comments <grin>.

<Jeff> Source not coming with comments?

<Beery> That's where it stands now.

<Beery> I know after talking with Lou that I brought the subject up and he said Paul had sparse comments.

<Jeff> Sounds like Paul.

<Clint> After staring at disassemblies for several years ANY kind of source code is an improvement!!

<Beery> Yes, most definitely.

<Dan> Do you get source to the DSR's?

<Beery> I think one of the most interesting pieces of Paul's code I have looked at was a disassembled, fully commented piece from the GPL interpreter. Dan, I don't have any MDOS source yet.

<Dan> When you get the source code, will Lou through in the source from the rest of myarc products.... DSR's and perhaps disk managers?

<Jeff> I would not bet on that.

<Beery> I don't know.

<Beery> I have the source for the floppy disk controller, myarc 512K, and I think the RS232 already. go

<Jeff> You probably have source to Personality card and not FDC.

<Beery> I've got both.

<Jeff> Interesting. Back to MDOS discussion. You mentioned Paul's GPL interpreter source. What made it so interesting?

<Beery> Essentially, the GPL interpreter is written to assemble at >=400 (for the assembly buffers), but once it is assembled, i've had to hand modify the load address for MDOS to be >0400. That's only the start. There are several pieces of code that get relocated all over the place to "handle" the onboard ram and memory mapping that it can almost get confusing. And then there are all the patches it makes into the OS to handle the extra 9995 features, etc. AND it all must be 7K, no larger without running into other problems. go

<Jeff> Sounds like a microcosm of the problems you'll encounter with MDOS.

<Beery> I think the majority of those problems if they appear will be either in the task switching area and/or possibly the DSR area. Those areas have to "remember" the onboard 9995 ram.

<Barry.B.> <-- would LOVE to alter the MDOS interrupt handler...

<Jeff> Maybe scrap and replace instead of overhaul, Barry.

<Beery> Barry, I'm sure you will have your chance as long as you implement that 38400 routine (grin), especially since I have a need for it now.

<Barry.B.> <grin> You should see my interrupt handler I did for my copy of Gentr... allows re-entrant interrupts :D

<Jeff> Beery, is there any hope that the Vid XOP's will have better comments because Paul did not originally write them?

<Beery> I have no idea.

<Jeff> Hi, Warren. I paged you because you mentioned some MDOS problems in the past.

BARNESW> Yes this was with the way MSDOS set the palette register via the XOP.

<Jeff> Maybe Clint can shed some light on that problem.

<Clint> Can you expand on the palette reg. problem?

BARNESW> The problem wasn't exactly how MDOS was setting the register, but how the two or three bytes necessary to define a color are arranged. I believe the proper sequence is the OP code is in

reg 0, reg 1 has two bytes to set the Red and Green hue and reg 2 has one byte in it to set the blue hue. According to the way the 9938 manual, four bytes are moved to the palette register that is going to be defined they are something like this >0000 >xx00 where xx = not defined. When setting the codes up in Regs 1 & 2, I used the same sequence, how ever the color was never defined properly. GA

- <Clint> OK, I'll look at it. Can you send this one to the "clearing house" too?
- <Barry.B.> <-- 9938 uses 4 nybbles to define a color...
- BARNESW> OK I'll do that along with the source code. My mistake - I meant nybbles!
- <Jeff> Warren said he did get it too work by skipping the XOP.
- BARNESW> Yes I did it via direct writes to the ports.ga
- <Jeff> R2=0GRB according to MDOS doc's.
- <Clint> I didn't think anything was wrong when I wrote the C palette function.
- <Beery> I would like to mention one thing.
- <Beery> With the docs that have been available on the video XDP's, we need to follow those latest releases (Volume 2 #5/3#1) as "gospel" as much as we can. Changing things outside of what is documented another way will cause problems with other programs. I'm not sure if the palette problem falls into that category.
- BARNESW> I'm using a very OLD version of the XOP manual, where can a newer version be obtained. As I stated before I was not sure my problem was not an MDOS problem, but rather one of a non-complete manual.
- <Beery> I had the latest compilation in 9648 News, Volume 3 #1 as I recall.
- <Jeff> Latest Vid XOP doc is in the 9648 database on Delphi.
- <Beery> I think the Vid 0.96H compilation on here by Al may have covered it.
- BARNESW> I'll check those out and compare to what I have. Other than the pseudo-palette problem I have seen no more.
- <Jeff> Last time we got in a heavy discussion about parent and child tasks. Is that something that needs further discussion? Or maybe a more pressing concern is if we can expect any help

- from Paul deciphering the source code once we have it.
- <Beery> Jeff, I think Al has written a pretty good set of C routines (unreleased) that has handled that on the programmer level. I think he can properly implement that into MDOS and solve the memory task problems that caused him to write those routines in the first place.
- <Barry.B.> <-- I am thinking that axeing the spooler is a good idea... then using an external "PRINT" command is Messy DOS
- <Barry.B.> would make the interrupt MUCH cleaner
- <Jeff> Good idea. I never can remember how to disable the spooler, and usually just turn off the printer.
- <Barry.B.> PRINT could be implemented as a time sliced task too... only side effect is a loss of spooling functions while in GPL
- <Jeff> Not if you make GPL a MDOS task.
- <Barry.B.> maybe EXEC could be built into MDOS if enough nonessential BS were eliminated
- <Jeff> But couldn't the Master DSR in GPL enable spooling capability?
- <Barry.B.> I suppose... I just hate to have the interrupt checking 5 zillion things.
- <Jeff> I.e., it only takes a TB instruction to determine MDOS/GPL mode. And obviously, the interrupt checking NEEDS to determine the mode. If not GPL, just skip checking for spooling.
- <Barry.B.> true... however the interrupt handling of MDOS/GPL is too Kludgy
- <Jeff> Only the GPL interrupt routine would need to check the mode.
- <Barry.B.> I actually disabled the MDOS interrupt completely in my Gentril loader ... replaced it with a simple interrupt manager... only has trouble if someone uses PIO
- <Clint> We are discussing the Geneva interrupt routines. Hi Al - glad you could make it from wherever you are
- ABEARD> Just glad to be home for a few days. Don't let me interrupt...
- <Jeff> On the discussion of interrupts, the 9648 does little better than the 4R.

<Barry.B.> Right Jeff... my ISR manager allows 38400 bps to work flawlessly... even faster speeds can be handled.

<Clint> For what it's worth, I ran the sieve benchmark with and without interrupts and the time difference was about 4%.

<Barry.B.> .t bogs down with 9902 and spooler handling

<Jeff> But were any devices requiring servicing?

<Clint> Yeah, I should try it with the spooler active.

<Barry.B.> disable spooling, and there is not even a need to make GPL access the MDOS ISR handler

<Jeff> And have a sprite moving.

<Barry.B.> The result is much peppier Terminal emulators, among other things

<Clint> Sorry folks, I have to go now. I'll catch the transcript later.

<Jeff> But can a TE run as a background task?

<Barry.B.> It could with a redesigned MDOS int handler... if the only recognized external interrupt were the 9902, MDOS could capture and store incoming characters, then feed them to a running program

<Beery> To handle background TE work while say downloading, that would be a major mdos revision.

<Barry.B.> it could never keep from losing chars while floppy access were going for example, but at least it would be easier to even WRITE a TE for MDOS

<Beery> It would not be something to come out of any initial debug stage, unless Barry had some quick method of doing it.

<Barry.B.> this is an EVENTUAL plan, not an immediate one... sort of a pet project

<Beery> Even with the HFDC and DMA Access, it would be a MAJOR undertaking due to MDOS's architecture and the File architecture. Something I am sure probably only one person could do Barry <grin>.

<Barry.B.> I don't foresee a background BBS or background downloading... just a friendlier environment for Term software

<Beery> I've got a quick question for those still left tonight... How many people use the SPOOLER, and how many people have BUGS with the Ir system when using it?

ABEARD> I don't have a printer on the system, so I don't use it.

<Barry.B.> I have mondo problems with it (which is why I want it GONE :)

<Beery> In my case, when another file is sent to it, it interrupts the printing file and starts immediately (printing the second file, that is).

<Jeff> I seldom do printouts. That's not a very green thing to do.

CAL47> I use the spooler extensively and have noticed no unbearable problems.

GREGRPH> Beery, I don't use the spooler. I thought that when 1.14 or .95 or .95 was released, there was no need for it. In fact, I thought it was disabled.

<Beery> I think you need to set the size of it Greg.

<Barry.B.> I'd rather use the printer's own RAM buffer, -OR- an MDOS PRINT program that would use time slices and a slash option to specify buffer size

CAL47> Make that in GPL mode only - GenTri

<Beery> It's fine if you have a single document, but multiple documents and it ALWAYS crashes out.

<Jeff> I think I have SPOOL 0 in my AUTOEXEC.

ABEARD> It seems if MDOS could multi-task well, just handle this as a background task.

<Barry.B.> the spooler is an interrupt nightmare... screws up too much other software (like speed to term software, etc)

<Jeff> The big QUESTION is can the Geneve multi-task well.

<Barry.B.> the answer: NOT AS SET UP UNDER MDOS! the reason! Every time slice, 128 bytes are moved out and in the >F000->F080 space if nothing else

<Jeff> I wonder if it could with any OS. Blame Clint for that.

GREGRPH> Beery, If I have to print multiple documents from say Gen-Tri, as soon as the screen comes back up, I can go through the steps to print another document and it will print when the first one is done. I lose nothing.

CAL47> Same here.

<Beery> Greg, maybe it is something hardware I have. I

use MDOS for the printing.

- <Barry.B.> if every task were assigned one high speed workspace by MDOS, then no memory swapping would be necessary <sigh> and swapping would have less overhead -- limit would be 4 tasks... pretty reasonable I think
- <Beery> But due to everyone programming differently, and everyone using multiple high speed workspaces, problems.
- <Barry.B.> unfortunately the existing software base would not be compatible :)
- <Jeff> I suggested A LONG TIME AGO that MDOS should have had a MM XOP that requested private on-chip RAM memory if that was necessary.
- <Barry.B.> I about fell over when I heard that MDOS has 64 NOV instructions in a row :) I would like to tinker with interrupts in such a way as to enhance what we have to work with, but keep compatibility
- <Jeff> And I think the dust had already settled on me by that time, Barry.
- <Beery> It might be something that could be added for a new "level" of software, but as Barry said, we must maintain current compatibility.
- BARNESW> Sounds like something microsoft said way back when...look at there problems :D
- <Jeff> I suggest an MM XOP that would mark on-chip RAM as GLOBAL, meaning that it would not have to be swapped in and out by any task that had fast memory marked that way. That way that current tasks would still have those 64 NOV instructions when swapping in and out. ga
- <Beery> Jeff, there may be another solution to what you are wanting.
- <Barry.B.> trouble is... any task using it would have to restore it before another slice expired
- ABEARD> OK. I've used the same method in shell, works nicely to start/stop tasks.
- <Jeff> Huh, Beery and Barry?
- <Beery> If the ability to turn on/off the scheduler is in place, then a "control" program (windows, game, etc), could have it's own scheduler and MM "xop" call.
- <Barry.B.> I am of the opinion that terminal software is a class of software that should not be multitasked, which would potentially mean that a

program COULD call an XOP to disable all other tasks, then re-enable them on exit without the need to be in LIM1 0 mode the whole time

- <Beery> This allows current programs to be compatible, and the next advancement of task management to work around the OS. This is where we need to equate bits and pieces of MDOS for MDOS control when necessary.
- <Barry.B.> <grin> just tossing ideas around... to see how many holes they land with :)
- <Jeff> What both of you are suggesting would complicate matters extremely more than what I suggested, IMHO.
- ABEARD> Multi-tasking is great with terminal software. When you are doing d'loads and the like it is convenient to switch to other operations.
- <Barry.B.> agreed A1, just NOT on the 9640 unless of course you know how to keep from losing characters during disk i/o, etc -- one of the things we "inherited" from the 4A <sigh>
- <Jeff> All I am suggesting is a very simple XOP that adds two bytes to a task-head.
- <Beery> Jeff, these two bytes do what?
- <Jeff> The first byte tells where in on-chip RAM the "private" on-chip RAM begin, and the second the number of bytes.
- <Barry.B.> that would reduce memory swapping
- <Jeff> Forgetting the decrementer and NMI vector. The task loader would preload those bytes to >0000 for all tasks.
- <Barry.B.> I still like the idea of letting a task disable all other tasks so as to maximize interrupts without losing them
- <Beery> That would be possible I think....
- <Jeff> If a task calls the special MM XOP and says that >F000-F03F is global, then those two bytes of the task-head are loaded with >4040.
- <Beery> Barry, only a few pieces of code would be required to do that, and still keep the "scheduler" and interrupt routine active.
- <Barry.B.> true
- <Beery> Store in page >80 a list of all task 0 pages. When you need to turn off other tasks, map in page >80, find out how many tasks are active, and write a "shut-down" byte into word >8100

offset into each task 0 page. This is what Windows does to turn all tasks off. It is the loader so it stores it's own list of task 0 pages.

<Barry.B.> or, set a flag somewhere which disables task swapping entirely, leaving only the current one active that could be checked with one ABS instruction

<Jeff> Kind of usurping MDOS command.

<Barry> Yes, someplace where the scheduler is active. I was just looking, looks like the crowd has died down and it's starting to get late here. Even later for some other folks on EST time.

<Barry.B.> if the task does a BLWP #0, the flag would be cleared assuming it forgot to clear it itself

<Jeff> Maybe we should summarize what happened tonight for the transcript first?

<Barry> It's at least allowing us to get together all at once and discuss some ideas, something we can't normally do.

<Barry.B.> yes, being able to bounce ideas around is useful, and will be MUCH more so when we have the source code in-hand

<Barry> I think it lacks some structure, but not having code, makes it difficult to have much structure.

<Jeff> We discussed quite a bit that is not really MDOS.

<Barry> Being able to discuss freely here, with others comments is useful. did we?

<Jeff> MDOS has a kludgy interrupt routine, kludgy spooler, and the solution seems to replace or disable both instead of fix.

<Barry.B.> death to the spooler! :)

<Barry> I'm not sure on dumping the spooler, perhaps a method of enabling/disabling it similar to the scheduler.

<Jeff> Maybe MDOS needs a disk spooler. :)

<Jeff> I'm not sure I like the idea of a task having the capability to disable the scheduler.

<Barry.B.> some things necessitate it... like a term program

<Jeff> Maybe the task loader should be such that MDOS knows to disable the scheduler when a specific task is loaded.

<Barry> Sounds like "alternative" file header bytes in addition to F/G, etc. <Jeff> I.e., when MDOS loads your TE, it know then that when the TE is active the scheduler must be deactivated. When the TE closes, the scheduler comes back on.

<Barry.B.> either that or let the TE do it itself via XOP... I see no difference -- night Barry!

<Jeff> Is that not a good idea, Barry? Perhaps F or G with the MSbit set would say this task needs all the resources of the computer.

<Barry.B.> yes... do you foresee any need to enable/disable the scheduler within a program... i.e. sometimes enable it and sometimes disable it, other than the obvious LIM1 0/2

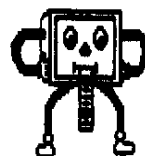
<Jeff> What I see as a potential problem is if two tasks want the scheduler off, there could possibly be problems with system resources. (not very good phrasing) A task should at least warn MDOS when it is loaded that it will require the scheduler off at times.

<Barry.B.> depends on why they want it off... a multi-level program like Gentr1 might find it useful to disable multitasking during TE mode, but enable it during Word processing -- ack it is getting late and my typing sloppier... I need to call it a night myself

<Jeff> Goodnight, Barry. Of course, we still have not solved the HFDC problems and have jumped to interrupts and multi-tasking.

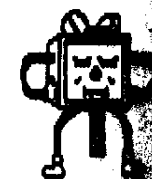
<Barry.B.> :D night!

--- END ---



VAPP
TO
PAGE PRO

BY DAN GAZSV



About a month ago, someone asked when there would be a conversion written to move VAPP pictures over to Page Pro. At the time, I didn't give it much thought. I have enough going on in my life without taking on the wants and wishes of every guy named Harry! The thought of moving VAPP pictures (and GIF through indirect methods) over to Page Pro, started to peak my interest. While I'd like to

Since just such a product exists, I can't and won't. I will do here is try to tell you everything I know about what is needed to accomplish this task.

First off, it's important to know the file spec of your file. Most of this is covered in the VAPP manual and addendum to the manual. Since there are two file formats available in VAPP (Mode 6 and 7), it's important to know both formats, they are different! The first two bytes of either file type is considered the header. The first byte denotes the backdrop color and the second byte is generally the value plugged into video register 8. This byte denotes the graphics mode of the file. Usually the only difference between the two file formats for this byte would be bit 2. Graphics mode 6 files would usually have a bit on the second byte, while mode 7 would usually have the value >FE. At this point, all the similarity between the file formats end.

Before going into the rest of the graphics 6 mode description, it might be helpful to talk about this mode and its capabilities. In mode 6, the width of the screen can hold 512 pixels x 212 pixels in length. In this mode, you can select 16 colors from a palette of a possible 512. If you have a palette to choose from, the VAPP file format will next contain the 16 palette colors that were chosen by the original author of the VAPP picture. The documentation in the VAPP manual addendum states that there are 16 bytes of pallet information after the header bytes, but that's wrong; there are 32 bytes (two bytes per palette entry). The format of the data is as follows: 1st nibble contains RED data, 2nd nibble contains BLUE data, 3rd nibble is not used, 4th nibble contains GREEN data. After the palette info, we have the picture data.

Picture data is in scan line format. That is, the data still represent color for the entire width of the screen before any info for the next scan line will be described. The picture data is stored in pairs of bytes. The most significant nibble (MSN) of the first byte denotes the palette color (something between >0 and >F). The rest of the two bytes (12 bits) denote the number of consecutive times that color should appear in the scan line. This pattern of byte paired data continues till the screen is completely painted.

Mode 7 pictures represent pictures that are 256 pixels (in width) x 212 pixels (in length). Unlike mode 6, there is no color palette in this mode, but you have access to 256 different colors. With this in mind, it would only make sense that picture data immediately follow the header bytes described earlier. In this mode, picture data also appears in byte pairs but the definition is a little different. The first byte denotes the color and the second byte denotes the number of consecutive bytes in that color. If the second byte should ever contain a count of zero, then it indicates that the remainder of the scan line is in the color selected by the first byte.

That's about all there is to decoding VAPP files. In my next installment, we'll discuss how to convert these color files over to black/white for printing purposes. In that installment we'll talk about things like dithering, threshold, artifacts, filters, error diffusion, serpentine scan and fixed dither patterns.



Hacking in c99



BY JOE ROSS

This month I will continue with the discussion of control statements such as "if", and "if/else". The usage of these controls are to test a variable/s in relation to a defined condition/s or value and depending on the outcome of the test, perform a certain task.

The syntax for the "if" and "if/else" are as follows:

```
if(expression/s) statement/s;
if(expression/s) statement/s; else statement/s;
```

In c99 you may have an "if" without an "else" but you may not have an "else" without a related "if". You may also nest them but be sure that you have paired the "if" and "else" expressions and statements correctly. If you desire to have more than one statement executed as a result of a particular "if" or "else", the statements must be grouped together within open and close braces. Another important aspect to mention about testing variables in c99 is that you make sure to use logical and not assignment symbols in the expressions.

Example:

```
if(a=1) /* This is not proper because the =
puts("A message"); /* single = sign will set "a" to 1 */
/* one and always test true. */
```

```
if(a==1) /* a proper test using == */
puts("a message");
```

The following are few examples of the if/if-else control.

```
-----
main()
{ int x;
  char c;
  x=0;
  while(x<10){ /* poll Keyboard ten times
    c=getchar(); /* get key value */
    if(c==48) /* if key is 0 perform next statement
      puts("You have pressed the zero key \n");
    else /* zero key not pressed
      puts("Not the zero key \n");
    x++; /* always increase x by one
  } /* go back and test is x<10
} /* end of program */
-----
```

The above program will poll the Keyboard ten times and

you to press a Key. If the Key is the zero Key it will tell you it in a message on the screen, else it will tell you it is not the zero Key.

```

-----
int()
int x;
char c,d;
d=0;
for(x=0; x<5; x++){
    c=getchar();           /* get Key from Keyboard */
    if(c>47 && c<58){      /* Is it 0-9 ? */
        puts("It's a number.\n"); /* YES */
        if(c==d){         /* Is it the same as last * */
            puts("Number same as last.\n"); /* YES */
            d=c;          /* make last * same as present */
        }
        else{             /* Not the same number */
            puts("Different number.\n");
            d=c;          /* make last * same as present */
        }
    }
    else{                 /* Key not a number */
        puts("Not a number.\n");
    }
}
/* end of "for loop" */
/* end of program */
-----

```

The above program will poll the Keyboard five times and each time wait for a Key to be pressed. If the Key pressed is 0 - 9 a message will appear on the screen telling that a number key has been pressed. If the number Key pressed is the same as the last number Key pressed then a message stating that it is the same number will be displayed else it will state that a different number has been pressed. If any key other than 0 - 9 was pressed a message stating that it is not a number will be displayed.

In the next article I will discuss the switch/case control statement.

Editor's Forum



How many times have you asked yourself, "When will the software take advantage of the hardware?" It seems that the hardware always arrives with a few software products (usually bundled by the hardware supplier) and then there is a long period of silence. The period of silence comes about for a number of reasons. The hardware supplier becomes disenchanted with sales of the product and usually directs his/her efforts to a different product line. This means that some of those who were initially in the product usually have to develop their own software for the product. If the individual is

resourceful and has a fairly good set of programming specs for the device, they can usually put out some nice software.

Some hardware vendors have the resources and forethought to provide the talented software developers a device with the stipulation that they develop software for the product. This has its good and bad points. The good points are: a) the software developer gets access to the device; b) the hardware vendor increases his potential for product sales based on the additional software that is available; c) the software usually gets to write/develop the software of their choice; d) the potential hardware owner doesn't get the very feeling that they are buying something that will just sit in a corner and collect dust.

Just as there are good points, there are also accompanying bad points too! a) most software developer's (for the 99/4a) have full-time jobs and only write software when they have time and a desire to do so; b) some software products are contracted, but the majority are the brainstorms of the software developer (suffice it to say, a dull programmer will most likely develop dull software); c) not all hardware is provided under the "open architecture" philosophy (some hardware vendors feel it necessary to protect their investments).

With that out of the way, let's get on to how most of the successful software arrives on the market. Knowing how small our market (99/4a and 9640) was and currently is, many of the more capable individuals were waiting for someone else to develop the products for their machine. After patiently waiting (for what seemed like a lifetime), many of us decided it was time to sit down and write the software ourselves. Some of the better products found their way to the market (via fairware or through third party vendors). While some found the fairware market to be successful, it was usually the exception rather than the rule. Within the next few months, there are promises of yet more hardware (SCSI device, 4a Menex, etc.) to arrive at our doorsteps. Hopefully, software will accompany it.

Joke Of The Month



"I hope the next time we see each other, Mary, it will be at the hospital," the obstetrician said to the pregnant patient.

"What would happen if I went into labor and I started to deliver the baby before I could get to the hospital?" Mary nervously asked.

"No problem. Childbirth is a completely natural event," he assured her. "Just lie in the same position as when you conceived and let nature take its course."

"You mean one foot in the glove compartment and the other out the window?"

99/4A - 9640**VENDORS**

Alpha Computer Supplies
 444 Hamilton Rd.
 Main Street Village
 Columbus, Ga 31909
 (706) 327-4222

Alpha Software
 P.O. Box 18386
 Columbia, Md 20850
 (301) 255-3885
 Catalog Available

Alpha Computer Services
 100 East Byrd Street
 Arlington, Va 22201
 (703) 731-3478
 (703) 731-4320 after 6pm
 Catalog #2

Alpha Hills Services
 100 Hartmouth Dr
 Columbus, Ohio 43214
 (614) 385-5946

Alpha Electronics
 50 Audubon Rd.
 Haverhill, Ma 01830
 (609) 895-0119

Alpha Computer
 2000 W National Ave
 Milwaukee, Wisc 53204
 (800) 242-7902 in Wisconsin
 (800) 662-9253 all others
 Catalog #1

Alpha Proline
 1945 Evergreen Ave.
 Fullerton, CA 92635
 (714) 990-4577 12-9pm PT

Alpha Computer Shopper
 P.O. Box F
 Titusville, Fl 32781

Alpha Comp
 111-G East Winston Rd.
 Anaheim, Ca 92806
 (714) 956-4450

Alpha Systems
 4345 Hortensia St
 San Diego, Ca 92109
 (619) 295-3301 voice
 (619) 270-8155 bbs

Alpha Only Software
 P.O. Box 244
 Lorton, Va 22079
 (310) 340-7179

Alpha Computerware
 835 Green Valley Dr.
 Philadelphia, Pa 19128
 (215) 483-1379

Alpha Lakes Software
 804 E. Grand River Ave.
 Howell, Mi 48843
 (517) 546-0566

Alpha Software
 5705 40th Place
 Hyattsville, Md 20781
 (301) 277-3467

Alpha Electronics
 4 N. 370 Pine Grove
 Bensenville, Il 60106
 (312) 766-9503

Alpha Inc.
 P.O. Box 23160
 Ft Orange, Fl 32029

Alpha Leshar
 722 Huntley
 Dallas, Tx 75214
 (214) 821-9274

Alpha Electronics
 P.O. Box 542546
 Dallas, Tx 75354-2526
 (800) 422-3892 in Texas
 (800) 527-7438 all others

Alpha Conner Enterprise
 1521 Ferry St.
 Lafayette, In 47904
 (317) 742-8146

Alpha Software
 P.O. Box 34160
 Omaha, Ne 68134

Alpha P.O.
 P.O. Box 1343
 Round Rock, Tx 78680
 (512) 255-1512

Alpha Engineering
 203 Arcadia Dr.
 Vernon Hills, Il 60061
 (312) 362-9834

Alpha Inc
 2624 Ranier Drive NE
 Birmingham, Al 35215
 (205) 854-5843

Alpha Polyoptics
 P.O. Box 4443
 Woodbridge, Va 22191
 (313) 499-5543

Alpha Pensive Abacutors OPA
 432 Jarvis Street
 Suite 502
 Toronto, Ontario
 CANADA M4V-2H3
 (416) 960-8925

Alpha Pride
 5 Williams Ln.
 Hatboro, Pa. 19040
 (215) 441-4262

Alpha Innovators
 4122 Glenway
 Mauwstose, WI 53222

Alpha 99 Software
 1884 Columbia RD #1821
 Washington, DC 20009
 (202) 667-3574

Alpha Computer Shoppe
 6102 Roosevelt Way NE
 Seattle, Wa 98115
 (206) 522-6558

Alpha Computers
 6467 E. Vancey Dr.
 Brookpark, Oh 44142
 (216) 243-1244 evenings

Alpha 99
 112 Rambling Road
 Vernon, Ct 06066
 (203) 871-7824

Alpha Ross
 119 Knollwood Terrace
 Clifton, NJ 07012

Alpha Tenex
 P.O. Box 6578
 South Bend, In 46660
 (800) 346-2778
 (217) 259-7051

Alpha Textaments
 53 Center St
 Patchogue, N.Y. 11772
 (516) 475-3480
 (516) 475-6463 bbs

Alpha Tex-Comp
 P.O. Box 33084
 Granada Hills, Ca 91344
 (818) 366-6631
 Catalog #2

Alpha The Bungalow Group
 P.O. Box 62323
 Colorado Springs, Co
 80962-2323
 (719) 488-2572

Alpha Tigercub Software
 156 Collingwood Ave.
 Columbus, Oh 43213
 (614) 235-3545

Alpha TM Direct Product Marketing
 1658 Broadway
 Redwood City, Ca 94063
 1-800-396-9966
 Catalog Available

Alpha Tricot Software
 P.O. Box 114-A
 Liscomb, Ia 50148

Alpha 9640 News
 c/o Beery Miller
 5455 Marina Cove #1
 Memphis, Tn 38115