

A SAMPLER OF THIS ISSUE: microPASCAL DOS Revealed : Variations on an Infinite Theme--Cursor Control : Using MicroMONITOR from the MeD

#### Have You Found Version 1.1 Software?

Try your local dealer. If, after reasonable time, he hasn't got it, write: Meizner Business Machines, Inc., 4771 Boston Post Road, Pelham, N.Y. 10803. For fast service, enclose a cashier's check or money order. Personal checks take two weeks to clear. Price: \$32.00 U.S. in the U.S. or Canada, including postage, packing, and handling. We got 1.1 from Meizner, after 1.5 months of waiting for our dealer. Commodore says distributors, not dealers, are the problem. Ask for Version 1.1 Software, with COBOL. It's all in plastic wrap, with manual updates.

#### A FORK IN THE ROAD

The Gazette started, purposely, with a new SPET just out of the box, and a beginner trying to make it work. We will sell, at cost (\$1.25, including postage, per issue), any back issues of the Gazette to new users. Write the editor and send your check. Our margins are set so each issue can be filed in a three-ring binder. Each volume is page-numbered consecutively. We need a volunteer to index them so we can issue a volume index. As time goes on, we should have a complete user's guide to SuperPET, and no need to repeat elementary material.

Each new issue then can and will present more advanced material. Starting with this issue, we've reached a fork in the road. We'll continue the elementary material until the basic machine is defined, but have started publishing more advanced stuff. See Barry Bogart's article on the MicroMONITOR, this issue.

#### ONCE OVER LIGHTLY--NOTES AND COMMENTS

Fred Fuller, Clinton, Iowa, writes that it's not a good idea to restrict contributions to disk in 8050 format (Vol. 1, No. 2). Fred, we accept contributions on dried intestine of goat or typed on toilet paper (unused). If you contribute on disk, put it in Word Pro or in the microEDITOR, used alone, 8050 format, and 80 characters wide. Fred is a radio ham, call NOFF, operates on all bands from 10 to 160 meters, and would like to hear from any SPUGGERS.

The enthusiasm out there has generated a lot of letters--unfortunately without a return envelope or postage. If you inquire or send material, send a self-addressed, postpaid return envelope. After the postage bill for last month, SPUG is even more NON-PROFIT than it began. And please print name and address. We have one inquiry we can read as five different addresses. It's not alone.

In this issue, we begin to explore SuperPET's powerful cursor control. The next page is a table of cursor positions on the screen, plus a summary of the graphics available from the shifted keypad, so listed you can find the symbol, chr\$ number, and function you want to use. Both tables are essential in planning cursor position, or in using SPET's graphics. (You can also get the old PET graphics, but that is another story.)

Row No. of Screen	Starting No. Screen Left	Mid-No. Mid-Screen	Ending No. Screen Right
1	1	40	80
2	81	120	160
3	161	200	240
4	241	280	320
5	321	360	400
6	401	440	480
7	481	520	560
8	561	600	640
9	641	680	720
10	721	760	800
11	801	840	880
12	881	920	960
13	961	1000	1040
14	1041	1080	1120
15	1121	1160	1200
16	1201	1240	1280
17	1281	1320	1360
18	1361	1400	1440
19	1441	1480	1520
20	1521	1560	1600
21	1601	1640	1680
22	1681	1720	1760
23	1761	1800	1840
24	1841	1880	1920
25	1921	1960	2000

TABLE 4-1

Cursor Positions on SuperPET

Example: Row 1, at the top of screen, starts with cursor position 1 (not 0); and ends at 80.

Line 2 (Row 2) starts at 81; mid-screen's 120, and end of line is 160.

The table makes plotting print positions a simple business.

Graphics Codes on Shifted Keypad

Category of Character	Example	Shifted Keypad Number	chr\$ number
Vertical Line		1	129
Horizontal Line		2	130
<b>Corners</b>			
Lower Right		3	131
Lower Left		4	132
Upper Left		5	133
Upper Right		6	134
<b>Intersections</b>			
Vertical Joins Bottom Horizontal		7	135
Left Horizontal Joins Vertical		8	136
Vertical Joins Top Horizontal		9	137
Right Horizontal Joins Vertical		0	138
Double Intersection		.	139

## Some Variations on an Infinite Theme--CURSOR CONTROL

Some SPET users say that SuperPET's cursor control isn't as responsive and easy to use as the old PET system. Perhaps, but many find it more responsive, easier to use, and more flexible. And you DO NOT have to rewrite your whole program to stuff in printable cursor commands for hard copy.

The problem lies in the manuals, which neglect cursor control. For example, the revised (Version 1.1) MicroBASIC manual, p. 202, prints three lines which don't tell you much.

Follow the manual. Ask the machine: ? cursor(10) in immediate mode and it prints 10 at column 11, row 1. If you say: print cursor(10);"100 turkeys", the screen shows "10 100 turkeys". You may read that as 10,100 turkeys, or you may scream and look for better ways. You'll find them. SuperPET has incredibly powerful cursor control, in two modes: absolute and relative. Each has its advantages.

## ABSOLUTE CONTROL

First, an "if" command is NOT conditional for the cursor. It is the "if" of an emperor: thou art commanded! Write: if cursor(1000) then print "Start printing at screen position 1000", and SPET will do exactly that.

The if...then statement is, however, long. It confines you to short print statements, or forces you to continue them on the next line. You can evade the full if...then:

1. With loops of all sorts: Example: You have an array of ten strings, each of less than 40 characters. You want a two column printout, starting on line 3. The array is named array\$. A little for...next loop does the trick with a single if...then:

```
20 a=161 ! initial cursor position, line 3
30 for i=0 to 9
40   if cursor(a) then print array$(i)
50   a = a + 40
60 next i
```

While loops will work, at both fixed and moving positions:

10 ! A fixed while loop	10 ! A moving while loop
20 while cursor(1860)	20 while cursor(i+161)
30   i = i+1	30   i=i+1
40   for j = 1 to 50	40   for j = 1 to 100
50   next j	50   next j
60   print "Time is: ";time\$	60 print " "; i
70 until i = 50	70 until i > 100

The j loops simply slow printing down enough for you to read it. You can print any variable instead of the simple time or i values above. While ...

endloops work just as well, using an "if...then...quit" statement to get out of the loop.

You can even tab with absolute cursor control, so long as you take into account the maximum length of the variable you want printed and the space needed between variables. Try this one:

```

10 ! Tabbing with absolute cursor control
20 i = 1
30 while cursor(i + 160)
40   i = i + 4
50   for j = 1 to 50
60     next j
70   print i
80 until i > 560

```

The method above works with either string or numeric variables. Be sure the increment of i (line 40) is one larger than your maximum variable length, and that 80 divided by the increment is going to get you back to the left margin. If you want, change the increment in line 40 to 5, and see what happens. Hope you like herringbone weave.

2. Here's a simple way to get rid of that long "if cursor(xxxx) then...." A short, simple procedure lets you print across almost the whole screen.

```

5 ! Maximum printing room with minimum typing.
10 b$="This should be on row 2, and we can run the statement to right margin"
20 b=81 : call pr
30 b$="This should be on row 4; again we can run all the way to the margin.."
40 b=241 : call pr
50 b$="This should be on row 14, and our print statement again runs fully..."
60 b=1041 : call pr
70 stop
80 !
90 proc pr
100  if cursor(b) then print b$
110  b=0 : b$="" ! Optional. Purist at work; cleans variables after last use.
120 endproc

```

You can relocate any line by simply changing the value of "b" (picked from table 4-1). The procedure slows printing about 10% from time taken with straight "if cursor....then" statements. We iterated two programs which printed the same lines 500 times to check. In normal use, you can't see the difference. Memory? Uses more. The first two lines of the program above use 12 more bytes than an if cursor...then statement to print the same line. Advantage of the method: easy to write. When print statements continue on the next line, you never are sure (unless you count characters) when your string is over screen width—and will split words and do other messy no-no's to your otherwise pleasing layout. The method can be useful.

## The Happy Change in Version 1.1

Page 110 of the revised MicroBASIC manual shows an example of the new multiple line statement. Here's an example with an if cursor...then:

```
10 if cursor(81) then print "This is version 1.1 method of continuing ";&
20 & "lines to next line." ! Semicolon on line 10 essential
```

## Maybe Some Fun and Games?

We've just skimmed absolute cursor control. We'll cover relative cursor control next issue; meanwhile, a little fun, if you care for it, with absolute control:

Follows a marquee (not a good, endless one, just a whumped up and bad one, purely for demo). The method it uses will take up to ten strings of 80 characters or less, in an array called b\$(j). It's set to repeat each string twice, but you can change that, when you find out why line 55 is set at 158, not 160.

To induce some thought, we offer a gold star (paper), publication, kudos, and a six-month extension to subscription to the Gazette to the writer of the best TRUE marquee program. Lines must scroll off the left of the screen as the marquee adds new ones to the right. The program must accept up to 10 strings, and allow any substitute strings of 80 characters or less. Marquee must be continuous, using all strings in sequence. Cleanest, shortest program wins. Contest ends with March/April issue, 1982; entry must be postmarked no later than February 1, 1983. Three judges (not in the contest). Use any SPET language; provide simple means and instructions for entry of substitute strings. Object: develop efficient screen and cursor control. Best from each language will be published.

```
10 ! cursor ex.10. A moving marquee. code 'marquee1'
15 print chr$(12): j=1
20 if cursor (661) then print "The marquee does not affect lines below."
25 if cursor (741) then print "When tired of this madness, press STOP"
30 b$(1)="Next Week, We Open the SuperPET Follies, Starring Walt Kutz "
35 b$(2)="Walt recites 'The Boy Stood on the Burning Deck', Sings, Dances "
40 b$(3)="The Editor Orates All 2098 Obscene Verses of 'Malemute Saloon' "
45 g=80-len(b$(j)) : g$=" ": d$=b$(j) + rpt$(g$,g)
50 i=80
55 while a < 158
60   k=k+1 : i=i-1 : a=a+1
65   if k=len(d$) then k=1
70   if cursor (i) then print str$(d$,1,k);
75   if i=1 and cursor(1) then print chr$(6)
80   if i=1 then i=80
85 until a = 158
90 j = j + 1
95 if j=4 then j=1
100 a=0 : k=0 : goto 45
105 end
```

Does what we've said on cursor control apply in other languages? Early on, Bob Davis (Associate in Pascal) noted that the ordinates listed for control in Vol. 1, No. 2 (chr\$(12) to clear screen, etc.) apply in languages other than microBASIC. The Editor suspects that what's said about cursor control in microBASIC applies, in some ways, to each of the other languages. When, next issue, we touch on relative cursor control, we're sure of more use across languages.

#### PASCAL DOS Commands

Bob Davis, bless him, has come up with the basic format for dos commands in PASCAL; he adds comments on "page" as a substitute for chr\$(12) to clear screen and to go to top of form on printer. Here's dos command format for PASCAL, editorial comments in (\* \*) which you may leave in program or take out.

```

program doscommand1 ( output);

var
    disk : text;

begin
    page;          (* clears screen; homes cursor *)
    writeln ( 'dos command issued' ); (* optional line to screen *)
    rewrite ( disk );
    writeln ( disk, '<enter dos command here>' (* < > are NOT used *)
end.

```

Save the form above; modify it to issue any dos command. Bob shows us an example of a rename command. We print only the change needed between begin...end to change a filename on disk 1 from doscommands to dos commands:

```

begin
    page;
    writeln ( 'Renaming file' ); (* optional line *)
    rewrite ( disk );
    writeln ( disk, 'R1:dos commands=doscommands' )
end.

```

The "page" command works equally well as a substitute for chr\$(12) if you go to the printer. We assume you have a Commodore 8300P (ASCII), IEEE bus, want to print what you've done, and then go to top of form, next printer page:

```

program doscommand2 ( output );

var
    disk, ieee4 : text;

begin
    page;
    writeln ( 'Renaming file' );
    rewrite ( disk );
    writeln ( disk, 'R1:dos commands=doscommands' );

```

```

rewrite ( ieee4 );
writeln ( ieee4, 'Renaming file doscommands to dos commands' );
page (ieeee4 )      (* printer goes to top of form *)
end.

```

So far, the Gazette has printed how to use the dos commands in all languages but APL and COBOL. ANYBODY out there who can issue dos commands from these languages, send us the short word on how!

Is it unreasonable to want to new a disk and save your experiments with a new language as you learn it, the first time you go to work? How to use dos and how to talk to disks and printer should be first page in every manual. It isn't.

#### On the ADA 1450 Interface and Switch Settings

If you use one, the official word is, switch 4 on, 1 through 3 off, with Diablo or Commodore 8300P printers. Off is switch down nearest the edge of the plastic case.

On the editor's SPET (and on others we know of), with switch 4 on, the printer capitalizes all lower case letters, and prints all lower case letters in caps (from 6809). One by one, trying all switch settings, we find (and have reported to us) only one combination that works in all modes and programs: SWITCH ONE ON, all others off. Please report anything different that works.

#### USING THE MicroMONITOR FROM THE MicroEDITOR

By Barry Bogart, 2405 West 15th Avenue, Vancouver, B.C., Canada, V6K 2Z1

(Ed. Note: Barry uses capitals to emphasize commands, though in the MeD they are entered in lower case. Enter Development from the Version 1.1 disk, and select edit mode. Thanks to Barry for a clear, well-illustrated article.)

You enter the monitor from the editor in version 1.1. (but probably not on the IBM PC) to use system subroutines from the MeD, giving it the PEEK, POKE, and SYS facilities of the other languages. I use it to invoke the TSETCHAR subroutine so I can edit with the APL character set. The principle should be the same for any use.

(Note: Be careful with the monitor, as with POKES; be sure the program you enter (with MODIFY) looks reasonable; disassemble it with the TRACE command. Only if it looks ok do you run it with GO.)

Here's an example, a program which changes the character set to APL while in the microEDITOR:

First, type MON to get into the monitor. Then type at the > prompt, at the bottom of the screen:

```
m 2000 cc 00 02 ed 8a bd d5 21 3f [Note: we'll call this line ENTRY]
```

You can verify the code, using TRACE, by entering: t 2000-2009. This will

display:

```

2000 LDD #$0002
2003 STD ,X
2005 JSR $d521
2008 SWI

```

This simple program stores '2' and jumps to the subroutine at \$d521. (Normal entry address for TSETCHAR\_ is \$b08a, but it's only a vector to \$d521. Why bother?)

To run it, type: g 2000 After it runs, you will see the uppercase line below in APL characters. You will also see INTERRUPT. All ok. You've got the APL character set. The subroutine ends with a software interrupt (SWI), not a return from subroutine (RTS). You must do this to get back to the MeD, even though it causes a register dump.

```

PC      D      X      Y      U      S      CC      DP
(2008   000E   0000   0000   0000   0220   C0      00

```

To return to the MeD from the monitor, type at >: q

To get back ASCII characters, reverse the process: go back to the line called ENTRY, above. Put in: m 2000 cc 00 01.... (Note we're storing a 1, not 2).

Here's a short-cut back (if the code in 2000 hasn't been clobbered): Use DUMP to modify memory directly. Enter the MONITOR with MON (caps, in APL). Type: D 2000+2009 to dump the code. Locate that 02, overtype with an 01, hit RETURN, and then do the G 2000 at the > to rerun. You're back in ASCII. (Remember the + sign, on the APL keyboard, is at the : on the regular keyboard.)

ADVICE NEEDED! ADVICE NEEDED!

Bernard Eisenberg, 26 Lourae Drive, Massapequa Park, N.Y. 11762, writes: "Since getting the Spinwriter, I have been unable print APL. I'm using an APL thimble, but it prints out the corresponding capital letters instead of the APL characters." Send the solution to Bernard directly (and send us a copy).

#### Comments on the Terminal Emulation Package

Dan Helman, 9 Mark Street, Boston, MA, 02130, says of Waterloo's Terminal Emulation package: "It is much better than SETUP and the TALK command in the editor, but it leaves something to be desired. The best features are its ability to copy to a file, and the addition of real control and escape keys to the keyboard." (Ed. note: Don Gilbreath of Dallas, one of our members, wrote Waterloo on the same package last summer and asked that the Escape function be moved to the Escape key. Waterloo updated for Don. This confirms our experience with Waterloo: they respond.) Dan continues, "...the flaws of the program are a faulty keyboard debouncing routine, and emulating an offbeat terminal (for my work). The keyboard often makes mistakes when you release the shift key and are

typing fast. For example, Hello comes out Hhello.... Wish they had emulated the DEC VT-100, which has ANSI Standard Codes." Dan would like to write his own emulation program, but allows he could use some help to figure interrupts and the keyboard in SPET.

APL Character Set Discovered by Accident

R.D. Connely, 424 South Florida Ave., Joplin, MO, 64801, writes: "Many BASE 2 800 mst dot matrix printers have the APL character set built in ROM. Although not documented, I discovered it before I ever heard of APL. It was generated by typing: open 1,4 : print#1, chr\$(27);chr\$(78)."

WE ACKNOWLEDGE YOUR CONTRIBUTIONS

They have rolled in, and we bless you for them--more than we can print this issue. Terry Peterson, 8628 Edgehill Court, El Cerrito, CA 94530, gives us a new and better screen dump (next issue), and programs to screen dump SPET graphics to an MX-80 printer (next issue). Terry notes that when you do chr\$(i + 128) for reverse printing to screen, you generate italics on an MX-80 with Graphtrax.

Allan Ackerman of 3678 Nicole, Las Vegas, Nevada 89120 contributes a BASIC 4.0 program which lets him to store 13,104 floating point numbers in the upper 64, which can be retrieved instanter. He uses it for a graphics routine which plots a bar chart with 3 simultaneous moving averages. It's 3.5 pages long in assembly language; we don't have room to print it. If you want a copy, send a self-addressed, stamped envelope and two 20-cent stamps to the editor (we pay 10 cents a page to the local Xerox). And send thanks to Allan.

Roy Busdiecker, PO Box 1442, Woodbridge, VA 22193 comes up with a beauty of an article on using the upper 64 from BASIC 4.0, and on stuffing POWER, COMMAND-O, or SM-KIT up there, (all or any) for instant use. And a program for loading. Next issue, we hope. Gary Ratliff has some basic stuff coming on assembler in 6502 and 6809, and we've got goodies from England, plus two public domain communications disks from Walt Kutz, which left Commodore Nov. 24 (no chance for a hard look yet).

AND, we met and exceeded our membership quota for the year. We can publish with assurance for at least one year, thanks to the readers. BANZAI!

\*\*\*\*\*

APPLICATION FOR MEMBERSHIP, SUPERPET USER'S GROUP

Name: \_\_\_\_\_ Disk Drive: \_\_\_\_\_ Printer: \_\_\_\_\_

Address: \_\_\_\_\_  
Street, PO Box City or Town State/Province/Country Postal ID#

For U.S. & Canada: Enclose Annual Dues of \$10:00 (U.S.) by check or money order, made out to Secretary, SPUG. Overseas dues: \$20.00 U.S. Mail to: Paul V. Skipski, Secretary, SuperPET User's Group, 4782 Boston Post Road, Pelham, N.Y. 10803, USA.

Newsletter published by the SuperPET User's Group (SPUG): editorial offices at PO Box 411, Hatteras, N.C. 27943. Secretary, Paul V. Skipski, 4782 Boston Post Road, Pelham, N.Y. 10803. Membership applications and inquiries to Mr. Skipski. All newsletter material to Hatteras, attn: Dick Barnes, Editor. SuperPET is a trademark of Commodore Business Machines, Inc. Contents of this newsletter copyrighted by SPUG, 1982; reprinting by permission only. SPUG members are authorized to use the material. Enclose a self-addressed, postpaid envelope with all material submitted and all inquiries requiring reply. Membership: \$10.00 per year, U.S. in North America, \$20.00 overseas and elsewhere. See enclosed application.

For all outside the U.S.: All nations members of the Postal Union offer certificates good in the postage of any other country for a small charge. The Union includes Canada, U.S., most European nations, Russia, China, and most of Araby. Each Gazette issue weighs one ounce: 20 cents U.S. & Canada, 80 cents to Europe. For other rates, see your local post office.

The SuperPET Gazette  
PO Box 411  
Hatteras, N.C. 27943

