



The fifth in a series of tutorials for the beginner to intermediate machine language programmer

# Machine Language Made BASIC

## Part V: Get the Point

By William P. Nee

This month's programs change the color of various screen locations. Now, color information is stored in several locations in the Color Computer. Locations used in this article are as follows:

Location		Start-up
\$B2	foreground color	(3)
\$B3	background color	(0)
\$B4	current color	(0)
\$B5	\$B4x(#\$55)	(0)
\$C2	PSET = 1; PRESET = 0	

As shown in Figure 1, colors depend on the PMODE and color set used.

The computer colors zero to three correspond to the BASIC colors one to four and five to eight. Three is the

or load the desired color number into Register B and JSR \$9536. (The latter will *not* change the background color in \$B3.)

Location \$C2 can be used as a toggle for PSET (if it is set to one) or to PRESET (if it is set to zero). However, the PSET routine we will use starts after the ROM routine has checked Location \$C2, so we should use either the subroutine at \$959A, which multiplies the color number by #\$55 and stores this total at Location \$B5, or PMODE 4/2/0, which will store a random 0 or -1 (RND(2)-2) in \$B5 and then PSET or PRESET the point. (Remember, -1 is the same number as #\$FF.) PMODE 3/1 will use a random (0 - 3)x(#\$55) to PSET three colors or PRESET the background color.

may have different branches if Register B was equal/not equal to one of these colors. If a point has been PRESET, the PPOINT routine will make it the background color. (Note: You always lose registers A, B, and X, so be sure to save them first.)

The PSET routine is at Address \$9374, and, as with most machine language programs, requires some additional set-up. The routine uses the following locations:

\$B9	bytes per line
\$BD/BE	horizontal coordinate — X1
\$BF/C0	vertical coordinate — Y1

The coordinate locations are two bytes so that you could, for example, either STX \$BD or STA \$BE. X1 cannot be greater than 255; Y1 cannot be greater than 191; and neither can be less than zero.

A scaling routine at \$931D is also required. Since we will pick up the PSET routine after ROM has scaled the coordinates, we must add this to our program prior to the PSET. Scaling adjusts X1 and Y1 to compensate for the different bytes per line (in \$B9) assigned to the individual PMODES. Without this scaling routine, most graphic commands (PSET, LINE, CIRCLE, etc.) would be accurate only in PMODE 4. (The PPOINT routine we've already discussed includes the scaling subroutine.)

Figure 1:

	Number	Set 0	Set 1	\$B5
PMODE 3/1	0	green	buff	#\$00
	1	yellow	cyan	#\$55
	2	blue	magenta	#\$AA
	3	red	orange	#\$FF
PMODE 4/2/0	0	black	black	#\$00
	3	green	buff	#\$FF

highest number used for color because 4x(#\$55) would be greater than 255 and would not fit into Location \$B5. With machine language, we can control and change the contents of the color locations throughout the program.

At start-up, the computer will store 0 in \$B3 (for the background color) and 3 in \$B2 (for the foreground color). Using the PCLS routine at \$9542 will clear the screen to the background color. If you want a different background color, either load the desired color number into \$B3 and JSR 9542,

You may also use the PPOINT routine at \$933C to check the color of a bit at any horizontal location (by storing that bit in Location \$BE) and any vertical location (by storing it in \$C0). The result of the PPOINT routine is stored in FP1. JSR \$B3ED will return the color number to Register B. In PMODE 3/1 the result will be colors one to four (if you are using color set to 0), or colors five to eight (if you are using color set to 1). In PMODE 4/2/0, the result will be zero or one with color set to 0, and zero or five with color set to 1. Your program

*Bill Nee bucked the "snowbird" trend by retiring to Wisconsin from a banking career in Florida. He spends the long, cold winters writing programs for his CoCo.*

**Listing 1: POINTBAS**

```

10 PMODE 4,1:PCLS:SCREEN 1,1
20 FOR N=0 TO 32
30 FOR NN=31 TO N STEP -1
40 B=RND(2)
50 FOR H=0 TO 192 STEP 64
60 FOR V=0 TO 128 STEP 64
70 ON B GOSUB 100,150
80 NEXT V,H,NN,N
90 GOTO 20
100 PSET(H+NN,V+NN-N):PSET(H+NN-
N,V+NN)
110 PSET(H+62-NN,V+NN-N):PSET(H+
62-NN+N,V+NN)
    
```

```

120 PSET(H+62-NN,V+62-NN+N):PSET
(H+62-NN+N,V+62-NN)
130 PSET(H+NN,V+62-NN+N):PSET(H+
NN-N,V+62-NN)
140 RETURN
150 PRESET(H+NN,V+NN-N):PRESET(H
+NN-N,V+NN)
160 PRESET(H+62-NN,V+NN-N):PRESE
T(H+62-NN+N,V+NN)
170 PRESET(H+62-NN,V+62-NN+N):PR
ESET(H+62-NN+N,V+62-NN)
180 PRESET(H+NN,V+62-NN+N):PRESE
T(H+NN-N,V+62-NN)
190 RETURN
    
```

The PSET routine in ROM uses registers A and B, so be sure to save any information in them first. Before running the program, set Location \$FF/100 to #2000. Since the program starts with PMODE, type "GPMODE" or "G3000" to execute it. Pressing any key will break the program, but you must hold the key down for several seconds because the program does a lot before getting to the break location. In the 'A' mode you can read the program and symbols from \$2200 to \$2B65.

Run the BASIC program first to get a feel for the design and program speed

(or lack of it). Next, run the machine language program. (Note: The machine language program does *not* run on the CoCo 3.) If you run the machine language program from BASIC, clear enough memory with the command, CLEAR 200, &H3000-1. Even though

the machine language program is eight times longer, it runs much more quickly.

(Questions or comments concerning this tutorial may be addressed to the author at Route 2, Box 216C, Mason, WI 54856-9302. Please enclose an SASE when requesting a reply.) □

**Listing 2: POINTBIN**

```

3000          00050 * $FF/100=#2000
              00100      ORG      $3000
              00110 PSET      EQU      $9374
              00120 X1       EQU      $BE
              00130 Y1       EQU      $C0
    
```

*Alpha Software Technologies*

**OS9 Level II BBS Release 2**

The best BBS system has just gotten better! System comes complete and ready to run in less than 5 minutes! Use the built-in menus or create your own, you can even run your own programs or games on-line! Complete message system allows easy message posting and retrieving. Complete file transfer system supports Xmodem and Ymodem as well as keyword searching, single line, and paragraph file descriptions; System runs completely in the background, allowing you full use of your computer.



512k OS9 Level II Required.....\$29.95



**The Zapper**  
This wonderful utility allows you to patch anything! Patch commands directly on the disk and fix CRC's automatically! Even allows you to patch the OS9boot file without making a new boot disk! Save files that have been lost or deleted! Fix crashed disks! Hundreds of uses!  
64k OS9 Level I or II Required.....\$19.95

**Disk Manager Tree**  
This versatile utility will make your OS9 life a breeze! No more will you have to fight with complicated directory structures. No more searching for files and typing long path names. All of this is displayed with windows. A tree window allows you to change, create, and delete directories quickly. A files window allows you to copy, view and delete files easily. Perfect for the OS9 beginner! Multi-View compatibility makes it perfect for Multi-View users!  
512k OS9 Level II Required.....\$29.95



**OS9 Toolkit**

Finally OS9 level becomes easy! With these great utilities you'll be using OS9 like a pro! Complete wildcard commands make file manipulation simple and easy! Tree commands make directory manipulation a breeze! The many other utilities make OS9 much easier to use! All utilities in all! At \$19.95 that's less than \$1.25 a utility!  
OS9 64k Level I or II Required.....\$19.95

**Level II Tools**

Level II tools includes all tools in the level I toolkit, plus 3 great level II utilities! Utilities include commands for changing colors, creating overlay windows, complete with borders, browsing through files, and more! At \$24.95 it's less than \$1.00 per utility!  
128k OS9 Level II Required.....\$24.95

**Warpl**

Finally, a complete OS9 level II terminal program that you can afford! Program features Auto-dial, Auto-macro, File transfers, buffer capture, on-line time, and much much more! Level II windows make it super simple to use! All you'll ever need in a terminal program! Comes complete with ICONS for Multi-View compatibility!  
512k OS9 Level II Required.....\$34.95

**Multi-Menu**

Easily create your own pop-down menus with this great utility! No programming experience necessary! With this great utility you can run any OS9 command or program from a menu! Menu creation is super-easy and super-simple! Actually create the menu from the menu! A must for any Multi-View user!  
512k OS9 Level II and Multi-View Required.....\$19.95

Please add \$3.00 for shipping and handling.  
C.O.D. orders please add an additional \$2.00 c.o.d. charge

Send check or money order to: Alpha Software Technologies  
2819 Buffon St.  
Chattanooga, La. 70043

C.O.D. orders call: (601) 266-2773

**Burke & Burke**   
P.O. Box 1283 Palatine, IL 60078-1283  
(312) 397-2898

**Real BASIC for OS9!**

OS-9 LEVEL TWO VR-... **Only \$39.95** ...  
COPYRIGHT 1986 BY MICROWARE SYSTEMS CORP. LICENSED TO TANDY CORP. ALL RIGHTS RESERVED.  
July 11, 1988 14:37:30  
Shell  
OS9: xmode /w5 type=0  
OS9: lnz /w5  
OS9: rab <>/w5 & &007  
CLEAR  
OK  
LOAD "DEMO"  
OK  
LIST  
10 PMODE 4:SCREEN 1,1  
20 X=RND(256)-1:Y=RND(192)-1  
30 A=RND(256-X)-1:B=RND(192-Y)-1  
40 LINE (X,Y)-(X+A,Y+B),PSET,BF

*There is nothing wrong with your Color Computer. Do not attempt to adjust it.*

Burke & Burke's new R.S.B. program gives you a complete, OS9-compatible version of Disk Extended Color BASIC. We've added new software for OS9-style graphics, sound, printer, and disk I/O. The BASIC you know and love is now running under Level 2 OS9 windows!

R.S.B. loads and saves files using OS9's file format, so we've also included utilities to transfer BASIC programs and data files between OS9 and BASIC disks. Of course, you can't use R.S.B. to run machine language programs, and some BASIC commands work slightly differently under R.S.B.

Your BASIC programs can take full advantage of great OS9 features like hard disks, no-hall floppies, multi-tasking, and 2 MHz operation.

R.S.B. requires a CoCo 3 with at least 128K RAM, a floppy controller with either Tandy Disk Extended Color BASIC or DISTO CoCo 3 CDOS ROM, and Level 2 OS9.

**R.S.B. \$39.95**

ILLINOIS RESIDENTS PLEASE ADD 7% SALES TAX. COD's add \$2.20. Shipping (within the U.S.) \$1.50. Please allow 2 weeks for delivery. Overnight delivery available for in-stock items. We accept MasterCard and VISA. Telephone orders accepted (312) 397-2898.

# Submitting Material To Rainbow

Contributions to THE RAINBOW are welcome from everyone. We like to run a variety of programs that are useful/helpful/fun for other CoCo owners.

**WHAT TO WRITE:** We are interested in what you may wish to tell our readers. We accept for consideration anything that is well-written and has a practical application for the Tandy Color Computer. If it interests you, it will probably interest lots of others. However, we vastly prefer articles with accompanying programs which can be entered and run. The more unique the idea, the more the appeal. We have a continuing need for short articles with short listings. These are especially appealing to our many beginners.

**FORMAT:** Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs and debug our typing errors. All programs should be supported by some editorial commentary explaining how the program works. We also prefer that editorial copy be included on the tape or disk using any of the word processors currently available for the Color Computer. Also, please include a double-spaced printout of your editorial material and program listing. Do not send text in all capital letters; use upper- and lowercase.

**COMPENSATION:** We do pay for submissions, based on a number of criteria. Those wishing remuneration should so state when making submissions.

For the benefit of those who wish more detailed information on making submissions, please send a self-addressed, stamped envelope (SASE) to: Submission Guidelines, THE RAINBOW, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit material *currently submitted* to another publication.

```

BF1F      00140 RND      EQU      $BF1F
3000 C6   04      00150 PMODE   LDB      #4
3002 BD   9628   00160      JSR      $9628
3005 C6   01      00170      LDB      #1          PAGE 1
3007 BD   9653   00180      JSR      $9653
300A BD   9542   00190      JSR      $9542   PCLS
300D C6   01      00200      LDB      #1          GRAPHICS SCREEN
300F BD   95AA   00210      JSR      $95AA
3012 C6   01      00220      LDB      #1          COLOR SET 1
3014 BD   9682   00230      JSR      $9682
3017 4F      00240 START   CLRA
3018 B7   3187   00250 LOOP1   STA      N
301B C6   1F      00260      LDB      #31
301D F7   3188   00270 LOOP2   STB      NN
3020 C6   02      00280 RANDOM  LDB      #2
3022 BD   BC7C   00290      JSR      $BC7C   REGISTER B TO FPI
3025 BD   BF1F   00300      JSR      RND      GET RANDOM(2)
3028 BD   B3ED   00310      JSR      $B3ED   FPI TO REGISTER D
302B 83   0002   00320      SUBD     #2      GET -1 OR 0
302E D7   B5      00330      STB      $B5     -1=#FF=PSET; 0=PRESET
3030 B6   3188   00340 Q1      LDA      NN      POINT 1
3033 B7   3189   00350      STA      XX
3036 B6   3188   00360      LDA      NN
3039 B0    3187   00370      SUBA     N
303C B7   318A   00380      STA      YY
303F 17   00B0   00390      LBSR     SHOW1
3042 B6   3188   00400 QQ1     LDA      NN      POINT 2
3045 B0    3187   00410      SUBA     N
3048 B7   3189   00420      STA      XX
304B F6   3188   00430      LDB      NN
304E F7   318A   00440      STB      YY
3051 17   009E   00450      LBSR     SHOW1
3054 86   3E      00460 Q2      LDA      #62     POINT 3
3056 B0    3188   00470      SUBA     NN
3059 B7   3189   00480      STA      XX
305C B6   3188   00490      LDA      NN
305F B0    3187   00500      SUBA     N
3062 B7   318A   00510      STA      YY
3065 17   008A   00520      LBSR     SHOW1
3068 86   3E      00530 QQ2     LDA      #62     POINT 4
306A B0    3188   00540      SUBA     NN
306D BB   3187   00550      ADDA     N
3070 B7   3189   00560      STA      XX
3073 F6   3188   00570      LDB      NN
3076 F7   318A   00580      STB      YY
3079 17   0076   00590      LBSR     SHOW1
307C 86   3E      00600 Q3      LDA      #62     POINT 5
307E B0    3188   00610      SUBA     NN
3081 B7   3189   00620      STA      XX
3084 86   3E      00630      LDA      #62
3086 B0    3188   00640      SUBA     NN
3089 BB   3187   00650      ADDA     N
308C B7   318A   00660      STA      YY
308F 17   0060   00670      LBSR     SHOW1
3092 86   3E      00680 QQ3     LDA      #62     POINT 6
3094 B0    3188   00690      SUBA     NN
3097 BB   3187   00700      ADDA     N
309A B7   3189   00710      STA      XX
309D 86   3E      00720      LDA      #62
309F B0    3188   00730      SUBA     NN
30A2 B7   318A   00740      STA      YY
30A5 17   004A   00750      LBSR     SHOW1
30A8 F6   3188   00760 Q4      LDB      NN      POINT 7
30AB F7   3189   00770      STB      XX
30AE 86   3E      00780      LDA      #62
30B0 B0    3188   00790      SUBA     NN
30B3 BB   3187   00800      ADDA     N
30B6 B7   318A   00810      STA      YY
30B9 17   0036   00820      LBSR     SHOW1
30BC B6   3188   00830 QQ4     LDA      NN      POINT 8
30BF B0    3187   00840      SUBA     N
30C2 B7   3189   00850      STA      XX
30C5 86   3E      00860      LDA      #62
30C7 B0    3188   00870      SUBA     NN
30CA B7   318A   00880      STA      YY
30CD 17   0022   00890      LBSR     SHOW1
30D0 F6   3188   00900 FINISH  LDB      NN

```

30D3	5A		00910	DECB	
30D4	F1	3187	00920	CMPB	N
30D7	102C	FF42	00930	LBGE	LOOP2
30DB	B6	3187	00940	LDA	N
30DE	4C		00950	INCA	
30DF	81	20	00960	CMPA	#32
30E1	1025	FF33	00970	LBLO	LOOP1
30E5	AD	9F A000	00980	JSR	[\$A000] ANY INPUT?
30E9	1027	FF2A	00990	LBEQ	START
30ED	5F		01000	CLRB	SET FOR TEXT SCREEN
30EE	BD	95AA	01010	JSR	\$95AA
30F1	3F		01020	SWI	RTS IF IN BASIC
30F2	FC	3189	01030	LDD	XX SQUARE 1
30F5	97	BE	01040	STA	X1
30F7	D7	C0	01050	STB	Y1
30F9	BD	9374	01060	JSR	PSET
30FC	FC	3189	01070	LDD	XX SQUARE 2
30FF	8B	40	01080	ADDA	#\$40
3101	97	BE	01090	STA	X1
3103	D7	C0	01100	STB	Y1
3105	BD	9374	01110	JSR	PSET
3108	FC	3189	01120	LDD	XX SQUARE 3
310B	8B	80	01130	ADDA	#\$80
310D	97	BE	01140	STA	X1
310F	D7	C0	01150	STB	Y1
3111	BD	9374	01160	JSR	PSET
3114	FC	3189	01170	LDD	XX SQUARE 4
3117	8B	C0	01180	ADDA	#\$C0
3119	97	BE	01190	STA	X1
311B	D7	C0	01200	STB	Y1
311D	BD	9374	01210	JSR	PSET
3120	FC	3189	01220	LDD	XX SQUARE 5
3123	CB	40	01230	ADDB	#\$40
3125	97	BE	01240	STA	X1
3127	D7	C0	01250	STB	Y1
3129	BD	9374	01260	JSR	PSET
312C	FC	3189	01270	LDD	XX SQUARE 6
312F	C3	4040	01280	ADDD	#\$4040
3132	97	BE	01290	STA	X1
3134	D7	C0	01300	STB	Y1
3136	BD	9374	01310	JSR	PSET
3139	FC	3189	01320	LDD	XX SQUARE 7
313C	C3	8040	01330	ADDD	#\$8040
313F	97	BE	01340	STA	X1
3141	D7	C0	01350	STB	Y1
3143	BD	9374	01360	JSR	PSET
3146	FC	3189	01370	LDD	XX SQUARE 8
3149	C3	C040	01380	ADDD	#\$C040
314C	97	BE	01390	STA	X1
314E	D7	C0	01400	STB	Y1
3150	BD	9374	01410	JSR	PSET
3153	FC	3189	01420	LDD	XX SQUARE 9
3156	CB	80	01430	ADDB	#\$80
3158	97	BE	01440	STA	X1
315A	D7	C0	01450	STB	Y1
315C	BD	9374	01460	JSR	PSET
315F	FC	3189	01470	LDD	XX SQUARE 10
3162	C3	4080	01480	ADDD	#\$4080
3165	97	BE	01490	STA	X1
3167	D7	C0	01500	STB	Y1
3169	BD	9374	01510	JSR	PSET
316C	FC	3189	01520	LDD	XX SQUARE 11
316F	C3	8080	01530	ADDD	#\$8080
3172	97	BE	01540	STA	X1
3174	D7	C0	01550	STB	Y1
3176	BD	9374	01560	JSR	PSET
3179	FC	3189	01570	LDD	XX SQUARE 12
317C	C3	C080	01580	ADDD	#\$C080
317F	97	BE	01590	STA	X1
3181	D7	C0	01600	STB	Y1
3183	BD	9374	01610	JSR	PSET
3186	39		01620	RTS	
3187			01630	N	RMB 1
3188			01640	NN	RMB 1
3189			01650	XX	RMB 1
318A			01660	YY	RMB 1
	3000		01670	END	PMODE

Hint . . .

### Cursor Controls

Memory location 63372 controls the blink rate of the cursor on the CoCo 3's 40- and 80-column text screens. Simply poke this location with any value from 0 to 255 to change the rate. The default value is 11. Lower values increase the blink rate while larger values make the cursor blink slower. If you want to stop the Hi-Res cursor from blinking altogether, enter POKE63381,0. To restart the blinking, enter POKE 63381,1.

*Ken Ostrer  
Vancouver, Washington*

Hint . . .

### HPRINT Shortened

When entering BASIC programs, I like to use the shorthand version of the PRINT command (typing a question mark instead of typing out PRINT). Unfortunately, if you try entering H? on the CoCo 3 (instead of HPRINT), you will get a syntax error upon running the program. To solve this problem, enter the entire listing using H?. Then save the listing in ASCII format and reload it. All H? commands will be changed to HPRINT.

*Carl England  
Calhoun, Georgia*

Hint . . .

### BASICally a Setup

In order to make using my CoCo a little easier, I saved a program on my utilities disk that sets the printer baud rate, drive selection and other parameters. To make things even simpler, I named the program \*.BAS. Now when I want to start working, I just enter RUN"\* and let the computer set itself up, much in the way an AUTOEXEC file works on MS-DOS systems.

*Harold Grumann  
Atlanta, Georgia*