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 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, 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.

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

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 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.)

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

```
1Ø PMODE 4,1:PCLS:SCREEN 1,1
2Ø FOR N=Ø TO 32
3Ø FOR NN=31 TO N STEP -1
4Ø B=RND(2)
5Ø FOR H=Ø TO 192 STEP 64
6Ø FOR V=Ø TO 128 STEP 64
7Ø ON B GOSUB 1ØØ,15Ø
8Ø NEXT V,H,NN,N
9Ø GOTO 2Ø
1ØØ PSET(H+NN,V+NN-N):PSET(H+NN-
N,V+NN)
11Ø PSET(H+62-NN,V+NN-N):PSET(H+
62-NN+N,V+NN)
12Ø PSET(H+62-NN,V+62-NN+N):PSET
(H+62-NN+N,V+62-NN)
13Ø PSET(H+NN,V+62-NN+N):PSET(H+
NN-N,V+62-NN)
14Ø RETURN
15Ø PRESET(H+NN,V+NN-N):PRESET(H
+NN-N,V+NN)
16Ø PRESET(H+62-NN,V+NN-N):PRESE
T(H+62-NN+N,V+NN)
17Ø PRESET(H+62-NN,V+62-NN+N):PR
ESET(H+62-NN+N,V+62-NN)
18Ø PRESET(H+NN,V+62-NN+N):PRESE
T(H+NN-N,V+62-NN)
19Ø 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

```
                    ØØØ5Ø *  $FF/1ØØ=#$2ØØØ
3ØØØ                ØØ1ØØ           ORG     $3ØØØ
          9374      ØØ11Ø PSET      EQU     $9374
          ØØBE      ØØ12Ø X1        EQU     $BE
          ØØCØ      ØØ13Ø Y1        EQU     $CØ
```

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    ØØ14Ø RND      EQU     $BF1F
3ØØØ C6         Ø4      ØØ15Ø PMODE    LDB     #4
3ØØ2 BD         9628    ØØ16Ø          JSR     $9628
3ØØ5 C6         Ø1      ØØ17Ø          LDB     #1       PAGE 1
3ØØ7 BD         9653    ØØ18Ø          JSR     $9653
3ØØA BD         9542    ØØ19Ø          JSR     $9542    PCLS
3ØØD C6         Ø1      ØØ2ØØ          LDB     #1       GRAPHICS SCREEN
3ØØF BD         95AA    ØØ21Ø          JSR     $95AA
3Ø12 C6         Ø1      ØØ22Ø          LDB     #1       COLOR SET 1
3Ø14 BD         9682    ØØ23Ø          JSR     $9682
3Ø17 4F                 ØØ24Ø START    CLRA
3Ø18 B7         3187    ØØ25Ø LOOP1    STA     N
3Ø1B C6         1F      ØØ26Ø          LDB     #31
3Ø1D F7         3188    ØØ27Ø LOOP2    STB     NN
3Ø2Ø C6         Ø2      ØØ28Ø RANDOM   LDB     #2
3Ø22 BD         BC7C    ØØ29Ø          JSR     $BC7C    REGISTER B TO FP1
3Ø25 BD         BF1F    ØØ3ØØ          JSR     RND      GET RANDOM(2)
3Ø28 BD         B3ED    ØØ31Ø          JSR     $B3ED    FP1 TO REGISTER D
3Ø2B 83         ØØØ2    ØØ32Ø          SUBD    #2       GET -1 OR Ø
3Ø2E D7         B5      ØØ33Ø          STB     $B5      -1=#$FF=PSET; Ø=PRESET
3Ø3Ø B6         3188    ØØ34Ø Q1       LDA     NN       POINT 1
3Ø33 B7         3189    ØØ35Ø          STA     XX
3Ø36 B6         3188    ØØ36Ø          LDA     NN
3Ø39 BØ         3187    ØØ37Ø          SUBA    N
3Ø3C B7         318A    ØØ38Ø          STA     YY
3Ø3F 17         ØØBØ    ØØ39Ø          LBSR    SHOW1
3Ø42 B6         3188    ØØ4ØØ QQ1      LDA     NN       POINT 2
3Ø45 BØ         3187    ØØ41Ø          SUBA    N
3Ø48 B7         3189    ØØ42Ø          STA     XX
3Ø4B F6         3188    ØØ43Ø          LDB     NN
3Ø4E F7         318A    ØØ44Ø          STB     YY
3Ø51 17         ØØ9E    ØØ45Ø          LBSR    SHOW1
3Ø54 86         3E      ØØ46Ø Q2       LDA     #62      POINT 3
3Ø56 BØ         3188    ØØ47Ø          SUBA    NN
3Ø59 B7         3189    ØØ48Ø          STA     XX
3Ø5C B6         3188    ØØ49Ø          LDA     NN
3Ø5F BØ         3187    ØØ5ØØ          SUBA    N
3Ø62 B7         318A    ØØ51Ø          STA     YY
3Ø65 17         ØØ8A    ØØ52Ø          LBSR    SHOW1
3Ø68 86         3E      ØØ53Ø QQ2      LDA     #62      POINT 4
3Ø6A BØ         3188    ØØ54Ø          SUBA    NN
3Ø6D BB         3187    ØØ55Ø          ADDA    N
3Ø7Ø B7         3189    ØØ56Ø          STA     XX
3Ø73 F6         3188    ØØ57Ø          LDB     NN
3Ø76 F7         318A    ØØ58Ø          STB     YY
3Ø79 17         ØØ76    ØØ59Ø          LBSR    SHOW1
3Ø7C 86         3E      ØØ6ØØ Q3       LDA     #62      POINT 5
3Ø7E BØ         3188    ØØ61Ø          SUBA    NN
3Ø81 B7         3189    ØØ62Ø          STA     XX
3Ø84 86         3E      ØØ63Ø          LDA     #62
3Ø86 BØ         3188    ØØ64Ø          SUBA    NN
3Ø89 BB         3187    ØØ65Ø          ADDA    N
3Ø8C B7         318A    ØØ66Ø          STA     YY
3Ø8F 17         ØØ6Ø    ØØ67Ø          LBSR    SHOW1
3Ø92 86         3E      ØØ68Ø QQ3      LDA     #62      POINT 6
3Ø94 BØ         3188    ØØ69Ø          SUBA    NN
3Ø97 BB         3187    ØØ7ØØ          ADDA    N
3Ø9A B7         3189    ØØ71Ø          STA     XX
3Ø9D 86         3E      ØØ72Ø          LDA     #62
3Ø9F BØ         3188    ØØ73Ø          SUBA    NN
3ØA2 B7         318A    ØØ74Ø          STA     YY
3ØA5 17         ØØ4A    ØØ75Ø          LBSR    SHOW1
3ØA8 F6         3188    ØØ76Ø Q4       LDB     NN       POINT 7
3ØAB F7         3189    ØØ77Ø          STB     XX
3ØAE 86         3E      ØØ78Ø          LDA     #62
3ØBØ BØ         3188    ØØ79Ø          SUBA    NN
3ØB3 BB         3187    ØØ8ØØ          ADDA    N
3ØB6 B7         318A    ØØ81Ø          STA     YY
3ØB9 17         ØØ36    ØØ82Ø          LBSR    SHOW1
3ØBC B6         3188    ØØ83Ø QQ4      LDA     NN       POINT 8
3ØBF BØ         3187    ØØ84Ø          SUBA    N
3ØC2 B7         3189    ØØ85Ø          STA     XX
3ØC5 86         3E      ØØ86Ø          LDA     #62
3ØC7 BØ         3188    ØØ87Ø          SUBA    NN
3ØCA B7         318A    ØØ88Ø          STA     YY
3ØCD 17         ØØ22    ØØ89Ø          LBSR    SHOW1
3ØDØ F6         3188    ØØ9ØØ FINISH   LDB     NN
```

```
3ØD3 5A          ØØ91Ø        DECB
3ØD4 F1   3187   ØØ92Ø        CMPB    N
3ØD7 1Ø2C FF42   ØØ93Ø        LBGE    LOOP2
3ØDB B6   3187   ØØ94Ø        LDA     N
3ØDE 4C          ØØ95Ø        INCA
3ØDF 81   2Ø     ØØ96Ø        CMPA    #32
3ØE1 1Ø25 FF33   ØØ97Ø        LBLO    LOOP1
3ØE5 AD   9F AØØØ ØØ98Ø        JSR     [$AØØØ] ANY INPUT?
3ØE9 1Ø27 FF2A   ØØ99Ø        LBEQ    START
3ØED 5F          Ø1ØØØ        CLRB            SET FOR TEXT SCREEN
3ØEE BD   95AA   Ø1Ø1Ø        JSR     $95AA
3ØF1 3F          Ø1Ø2Ø        SWI             RTS IF IN BASIC
3ØF2 FC   3189   Ø1Ø3Ø SHOW1  LDD     XX      SQUARE 1
3ØF5 97   BE     Ø1Ø4Ø        STA     X1
3ØF7 D7   CØ     Ø1Ø5Ø        STB     Y1
3ØF9 BD   9374   Ø1Ø6Ø        JSR     PSET
3ØFC FC   3189   Ø1Ø7Ø SHOW2  LDD     XX      SQUARE 2
3ØFF 8B   4Ø     Ø1Ø8Ø        ADDA    #$4Ø
31Ø1 97   BE     Ø1Ø9Ø        STA     X1
31Ø3 D7   CØ     Ø11ØØ        STB     Y1
31Ø5 BD   9374   Ø111Ø        JSR     PSET
31Ø8 FC   3189   Ø112Ø SHOW3  LDD     XX      SQUARE 3
31ØB 8B   8Ø     Ø113Ø        ADDA    #$8Ø
31ØD 97   BE     Ø114Ø        STA     X1
31ØF D7   CØ     Ø115Ø        STB     Y1
3111 BD   9374   Ø116Ø        JSR     PSET
3114 FC   3189   Ø117Ø SHOW4  LDD     XX      SQUARE 4
3117 8B   CØ     Ø118Ø        ADDA    #$CØ
3119 97   BE     Ø119Ø        STA     X1
311B D7   CØ     Ø12ØØ        STB     Y1
311D BD   9374   Ø121Ø        JSR     PSET
312Ø FC   3189   Ø122Ø SHOW5  LDD     XX      SQUARE 5
3123 CB   4Ø     Ø123Ø        ADDB    #$4Ø
3125 97   BE     Ø124Ø        STA     X1
3127 D7   CØ     Ø125Ø        STB     Y1
3129 BD   9374   Ø126Ø        JSR     PSET
312C FC   3189   Ø127Ø SHOW6  LDD     XX      SQUARE 6
312F C3   4Ø4Ø   Ø128Ø        ADDD    #$4Ø4Ø
3132 97   BE     Ø129Ø        STA     X1
3134 D7   CØ     Ø13ØØ        STB     Y1
3136 BD   9374   Ø131Ø        JSR     PSET
3139 FC   3189   Ø132Ø SHOW7  LDD     XX      SQUARE 7
313C C3   8Ø4Ø   Ø133Ø        ADDD    #$8Ø4Ø
313F 97   BE     Ø134Ø        STA     X1
3141 D7   CØ     Ø135Ø        STB     Y1
3143 BD   9374   Ø136Ø        JSR     PSET
3146 FC   3189   Ø137Ø SHOW8  LDD     XX      SQUARE 8
3149 C3   CØ4Ø   Ø138Ø        ADDD    #$CØ4Ø
314C 97   BE     Ø139Ø        STA     X1
314E D7   CØ     Ø14ØØ        STB     Y1
315Ø BD   9374   Ø141Ø        JSR     PSET
3153 FC   3189   Ø142Ø SHOW9  LDD     XX      SQUARE 9
3156 CB   8Ø     Ø143Ø        ADDB    #$8Ø
3158 97   BE     Ø144Ø        STA     X1
315A D7   CØ     Ø145Ø        STB     Y1
315C BD   9374   Ø146Ø        JSR     PSET
315F FC   3189   Ø147Ø SHOW1Ø LDD     XX      SQUARE 1Ø
3162 C3   4Ø8Ø   Ø148Ø        ADDD    #$4Ø8Ø
3165 97   BE     Ø149Ø        STA     X1
3167 D7   CØ     Ø15ØØ        STB     Y1
3169 BD   9374   Ø151Ø        JSR     PSET
316C FC   3189   Ø152Ø SHOW11 LDD     XX      SQUARE 11
316F C3   8Ø8Ø   Ø153Ø        ADDD    #$8Ø8Ø
3172 97   BE     Ø154Ø        STA     X1
3174 D7   CØ     Ø155Ø        STB     Y1
3176 BD   9374   Ø156Ø        JSR     PSET
3179 FC   3189   Ø157Ø SHOW12 LDD     XX      SQUARE 12
317C C3   CØ8Ø   Ø158Ø        ADDD    #$CØ8Ø
317F 97   BE     Ø159Ø        STA     X1
3181 D7   CØ     Ø16ØØ        STB     Y1
3183 BD   9374   Ø161Ø        JSR     PSET
3186 39          Ø162Ø        RTS
3187             Ø163Ø N      RMB     1
3188             Ø164Ø NN     RMB     1
3189             Ø165Ø XX     RMB     1
318A             Ø166Ø YY     RMB     1
         3ØØØ    Ø167Ø        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