



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

Machine Language Made BASIC

Part VI: I Draw the Line

By William P. Nee

This article will discuss and use a routine that executes the BASIC LINE command in machine language. The execution of this command will not be faster in machine language than it is in BASIC, since the LINE command itself is a machine language routine whether it is called from a BASIC or a machine language program.

The machine language *program*, however, will run more quickly than the BASIC program because all addition and subtraction functions use registers A and B rather than FP1 and FP2. Also, the computer does not have to parse (read each character of) the program to check for a wrong command or a syntax error.

Finding the point where a machine language routine stops reading the BASIC program and actually starts executing the command is one of the major difficulties in implementing ROM routines. Sometimes we need to add a small, but necessary, subroutine like the SCALE subroutine in last month's article, because we have skipped a portion of the ROM routine.

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.

	Subroutine	Description
\$959A	LDB \$B2	foreground color
	TST \$C2	1 = PSET; 0 = PRESET
	BNE \$95A2	branch if PSET
	LDB \$B3	background color
\$95A2	STB \$B4	color to use
	LDA #\$55	
	MUL	to fill one byte
	STB \$B5	color byte
	RTS	

Figure 1

I have included a subroutine at Address \$959A that uses the foreground or background colors to either PSET or PRESET. The subroutine is shown in Figure 1.

The TST command checks the CC register to see whether or not a number is equal to zero, and can also check to see if a signed number is positive or negative. You can then branch accordingly.

This routine works because PRESET works invisibly, like drawing a red line on a red background: It's there; you just can't see it. This subroutine was not part of the PSET routine we used last month, and it is not part of the LINE routine used in this article. We must skip the routine because the program looks at BASIC for the proper syntax during the ROM PSET and LINE routines. Our routine must start after this process.

If you are going to use a lot of color

changes and LINE PRESETs in your program, you may want to include this subroutine just before the LINE routine. However, you may find it easier to enter the proper color number (#\$0, #\$55, #SAA, or #SFF) into Location \$B5 prior to the LINE routine. Entering the background color number will PRESET; any other color number will PSET. See what happens if you put color number other than these in Location \$B5.

The BASIC LINE command requires an old x_1, y_1 and a new x_2, y_2 . In machine language, these coordinates are located at the following locations:

	Location
x_1	\$BD/BE
y_1	\$BF/C0
x_2	\$C3/C4
y_2	\$C5/C6

These locations are two bytes long, so your program could use registers X or

Y as well as registers A or B to load the locations.

Again, we must scale these locations or our program will work only in PMODE 4. We have already used the routine at \$931D to scale x_1, y_1 at \$BD and \$BF, but to scale x_2 and y_2 , we must load Stack U with the actual location of the x_2 coordinate and perform a JSR \$9320. The entire scaling routine is as follows:

```
JSR $931D      scales x1,y1
LDU #$C3      x2 starts at
               $C3
JSR $9320      scales x2,y2
```

Normally, this routine should be done just before executing the LINE routine at \$94A1. Anything necessary in registers A or B must be saved prior to using the scale routine.

The program in this article is in PMODE 3 and uses all available colors. It draws a line pattern in the following colors and order — cyan, magenta, orange and magenta — on a buff background (Color Set 1). To see Color Set 0, change the '1' in Line 190 to a '0'. Instead of using the subroutine at Address \$959A, the program puts the color number directly into Location \$B5 (lines 210, 420, 630 and 840). When

it has done this, press any key to interrupt execution of the program.

Before entering this program, set Location \$FF/100 to #2000. This moves the EDTASM+ buffer up to Location \$2000, well beyond graphics but below the ORG address of the program. If you run the program from BASIC, clear enough memory by using CLEAR 200, &H3000-1.

(Questions or comments concerning this tutorial may be directed to the author at Route 2, Box 216 C, Mason, WI 54846-9302. Please enclose an SASE when requesting a reply.) □

Listing 1: BASLINE

```
10 PMODE 3,1:PCLS:SCREEN 1,1
20 COLOR 6,5
30 FOR X=0 TO 92 STEP 4
40 LINE(220-X,96-X)-(128-X,4+X),
PSET
50 NEXT
60 COLOR 7,5
70 FOR X=0 TO 92 STEP 4
80 LINE(128-X,4+X)-(36+X,96+X),P
SET
90 NEXT
100 COLOR 8,5
110 FOR X=0 TO 92 STEP 4
120 LINE(36+X,96+X)-(128+X,188-X
),PSET
130 NEXT
140 COLOR 7,5
150 FOR X=0 TO 92 STEP 4
160 LINE(128+X,188-X)-(220-X,96-
X),PSET
170 NEXT
180 GOTO 180
```

Listing 2: BINLINE

```
00100 * $FF/100=#2000
3000 00110 ORG $3000
3000 C6 03 00120 START LDB #3
3002 BD 9628 00130 JSR $9628
3005 C6 01 00140 LDB #1 PAGE 1
3007 BD 9653 00150 JSR $9653
300A BD 9542 00160 JSR $9542 PCLS
300D C6 01 00170 LDB #1 GRAPHICS SCREEN
300F BD 95AA 00180 JSR $95AA
3012 C6 01 00190 LDB #1 COLOR SET 1
3014 BD 9682 00200 JSR $9682
3017 C6 55 00210 COLOR1 LDB #$55 CYAN
3019 D7 B5 00220 STB $B5
301B 5F 00230 LINE1 CLR B
301C 86 DC 00240 LOOP1 LDA #220
301E 17 00A7 00250 LBSR SUBT
3021 97 BE 00260 STA $BE X1 LOCATION
3023 86 60 00270 LDA #96
3025 17 00A0 00280 LBSR SUBT
3028 97 C0 00290 STA $C0 Y1 LOCATION
302A 86 80 00300 LDA #128
302C 17 0099 00310 LBSR SUBT
302F 97 C4 00320 STA $C4 X2 LOCATION
3031 86 04 00330 LDA #4
3033 17 0097 00340 LBSR ADN
3036 97 C6 00350 STA $C6 Y2 LOCATION
3038 34 04 00360 PSHS B
303A 17 0095 00370 LBSR LINE
```

303D	35	04	00380	PULS	B		
303F	CB	04	00390	ADDB	#4		
3041	C1	5C	00400	CMPB	#92	END OF LINE1 YET?	
3043	23	D7	00410	BLS	LOOP1		
3045	C6	AA	00420	COLOR2	LDB	#\$AA	MAGENTA
3047	D7	B5	00430	STB	\$B5		
3049	5F		00440	LINE2	CLRB		
304A	86	80	00450	LOOP2	LDA	#128	
304C	8D	7A	00460	BSR	SUBT		
304E	97	BE	00470	STA	\$BE	X1 LOCATION	
3050	86	04	00480	LDA	#4		
3052	8D	79	00490	BSR	ADN		
3054	97	C0	00500	STA	\$C0	Y1 LOCATION	
3056	86	24	00510	LDA	#36		
3058	8D	73	00520	BSR	ADN		
305A	97	C4	00530	STA	\$C4	X2 LOCATION	
305C	86	60	00540	LDA	#96		
305E	8D	6D	00550	BSR	ADN		
3060	97	C6	00560	STA	\$C6	Y2 LOCATION	
3062	34	04	00570	PSHS	B		
3064	8D	6C	00580	BSR	LINE		
3066	35	04	00590	PULS	B		
3068	CB	04	00600	ADDB	#4		
306A	C1	5C	00610	CMPB	#92	END OF LINE2 YET?	
306C	23	DC	00620	BLS	LOOP2		
306E	C6	FF	00630	COLOR3	LDB	#\$FF	ORANGE
3070	D7	B5	00640	STB	\$B5		
3072	5F		00650	LINE3	CLRB		
3073	86	24	00660	LOOP3	LDA	#36	
3075	8D	56	00670	BSR	ADN		
3077	97	BE	00680	STA	\$BE	X1 LOCATION	
3079	86	60	00690	LDA	#96		
307B	8D	50	00700	BSR	ADN		
307D	97	C0	00710	STA	\$C0	Y1 LOCATION	
307F	86	80	00720	LDA	#128		
3081	8D	4A	00730	BSR	ADN		
3083	97	C4	00740	STA	\$C4	X2 LOCATION	
3085	86	BC	00750	LDA	#188		
3087	8D	3F	00760	BSR	SUBT		
3089	97	C6	00770	STA	\$C6	Y2 LOCATION	
308B	34	04	00780	PSHS	B		
308D	8D	43	00790	BSR	LINE		
308F	35	04	00800	PULS	B		
3091	CB	04	00810	ADDB	#4		
3093	C1	5C	00820	CMPB	#92	END OF LINE3 YET?	
3095	23	DC	00830	BLS	LOOP3		
3097	C6	AA	00840	COLOR4	LDB	#\$AA	MAGENTA
3099	D7	B5	00850	STB	\$B5		
309B	5F		00860	LINE4	CLRB		
309C	86	80	00870	LOOP4	LDA	#128	
309E	8D	2D	00880	BSR	ADN		
30A0	97	BE	00890	STA	\$BE	X1 LOCATION	
30A2	86	BC	00900	LDA	#188		
30A4	8D	22	00910	BSR	SUBT		
30A6	97	C0	00920	STA	\$C0	Y1 LOCATION	
30A8	86	DC	00930	LDA	#220		
30AA	8D	1C	00940	BSR	SUBT		
30AC	97	C4	00950	STA	\$C4	X2 LOCATION	

30AE 86	60	00960	LDA	#96	
30B0 8D	16	00970	BSR	SUBT	
30B2 97	C6	00980	STA	\$C6	Y2 LOCATION
30B4 34	04	00990	PSHS	B	
30B6 8D	1A	01000	BSR	LINE	
30B8 35	04	01010	PULS	B	
30BA CB	04	01020	ADDB	#4	
30BC C1	5C	01030	GMPB	#92	END OF LINE4 YET?
30BE 23	DC	01040	BLS	LOOP4	
30C0 BD	ADFB	01050	JSR	\$ADFB	WAIT FOR INPUT
30C3 5F		01060	CLRB		TEXT SCREEN
30C4 BD	95AA	01070	JSR	\$95AA	
30C7 3F		01080	SWI		RTS IF IN BASIC
30C8 34	04	01090	PSHS	B	
30CA A0	E0	01100	SUBA	,S+	
30CC 39		01110	RTS		
30CD 34	04	01120	PSHS	B	
30CF AB	E0	01130	ADDA	,S+	
30D1 39		01140	RTS		
30D2 BD	931D	01150	JSR	\$931D	SCALE X1 AND Y1
30D5 CE	00C3	01160	LDU	#\$C3	START OF X2
30D8 BD	9320	01170	JSR	\$9320	SCALE X2 AND Y2
30DB BD	94A1	01180	JSR	\$94A1	DRAW THE LINE(X1, Y1)-(X2, Y2)
30DE 39		01190	RTS		
	3000	01200	END	START	

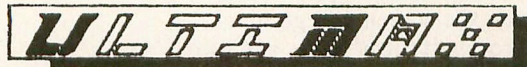
00000 TOTAL ERRORS

CocoTech NEW



With **MACPLAY** and a CoCo 3 you can play MAC sound files (Included on the second diskette) with pure 6 BIT sound quality that the CoCo can produce! Or download other MAC sound files from a computer information service or MAC bulletin board systems to hear even more. Sound files can last for a few seconds or up to 1/2 minute. MAC sound files included on the second diskette contain excerpts from the 3 Stooges to the Road Runner and more!!

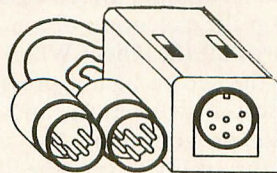
MACPLAY is only.....\$ 19.95



You can use **UltiMax** as a normal HI-RES joystick interface or switch it to be used with a popular Max III graphics program. The other feature of **UltiMax** is the option to have a large or small stick area so you can be more accurate with your drawings!

The **UltiMax** interface is only \$24.95

Or trade in your original HiRes interface (sent postage prepaid) and get **UltiMax** for only.....\$14.95



Send to:
CocoTech

PA residents add 6% sales tax
208 Cathy Ann Drive
Reading, PA 19606
(215)-779-7768

Please allow 1 to 3 weeks for delivery

Shipping and handling:
USA and Canada add \$2.50
Other countries add \$5.00

Sorry no credit cards
YET!

** We now handle C.O.D.'s **

JUBILEX

A fast paced arcade game that requires skill and quick thinking. Pilot your ship over the planet Jubilex. Avoid cannon shots and rockets from the ground while you battle their aircraft. You can modify and improve your weapon system. Requires joystick, CoCo III, and disk drive. \$25

GAT BACKUP

The ultimate 128k CoCo III backup utility. It gives you the options to backup the entire disk, a section, or only the granules used. It makes multiple copies, copies 35 tracks in two passes -- fast, and formats and gives directories. \$15

DIASM

This disassembler can disassemble files even if they overlap Diasm, Basic, or other dangerous parts of memory -- a must for hackers. It can insert, delete, and change bytes. Works with auto-executing programs. Supports printer. Many other features. CoCo I, II, or III. Disk only. \$20

OMNI UTILITY

The ultimate CoCo III disk utility. An on screen directory allows you to copy, kill, list, execute, and rename files at the touch of a key. Includes a full featured sector editor. Format, backup, and verify disks. Alphabetize and move directory entries. Many other features. \$20

All programs are 100% machine language. For an extra \$5 you can get the **source file** with the program. We pay shipping and sales tax. Write for more information, or send check or money order to:

GSW Software
8345 Glenwood
Overland Park, KS 66212