32K ECB

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

Machine Language Made BASIC

Part IV: Getting Graphic

By William P. Nee

With this article we start to explore the Color Computer's best capability: the speed and ease with which it can create graphics. The standard way to begin any graphics program is with PMODE, PCLS and SCREEN. These three commands, along with PCLEAR, will set certain locations within the memory. The main locations we will use for graphics and their meanings are shown in Figure 1.

Location	Indicates	Start	W/Disk
$B6	PMODE (1 - 4)	(0)	(0)
$B7/B8	end of graphics page +1	(#$C00)	(#$1400)
$B9	bytes per line (#$10 or #$20)	(#$10)	(#$10)
$BA/BB	start of graphics page	(#$600)	(#$E00)
$BC/BD	(#$600 if not disk, #$E00 if disk)		

Figure 1

When you first power up, the computer assumes PCLEAR 4, PMODE 0, PAGE 1, and sets the addresses as indicated under START. Since PMODE 0,1 uses only the first graphics page, the computer assumes that you will be using $600 to $C00-1 for graphics ($E00 to $1400-1 with disk). If not, you must tell the computer something different. The three main graphic commands in machine language are shown in Figure 2.

```
1) PMODE  LDB  #(0 - 4)   PMODE 0 to PMODE 4
          JSR  $9628
   PAGE   LDB  #(1 - 8)   PAGE 1 to PAGE 8
          JSR  $9653
2) PCLS   JSR  $9542      PCLS
3) SCREEN LDB  #(0 - 1)   graphics = 1, text = 0
          JSR  $95AA
          LDB  #(0 - 1)   color set 0 or color set 1
          JSR  $9682
```

Figure 2

Nothing is as easy as it looks. If you try these commands, your machine language program will become lost. The problem arises between the PCLS command and where *EDTASM+* stores the program. On power-up with *EDTASM+*, Location $FF/100 is #$600; this is where the edit buffer (your typed-in program) and the symbol table will begin. Since graphics also begin at $600, a PCLS will set all graphic bits to 0 and,

Address	Description	PCLEAR4	W/Disk	PCLEAR8	W/Disk
$19/1A	basic starts @	#$1E01	#$2601	#$3601	#$3E01
$1B/1C	variables start @	#$1E03	#$2603	#$3603	#$3E03
$1D/1E	arrays start @	#$1E03	#$2603	#$3603	#$3E03
$1F/20	free memory @	#$1E03	#$2603	#$3603	#$3E03
$33/34	data statements @	#$1E00	#$2600	#$3600	#$3E00
$A6/A7	input buffer @	#$1E00	#$2600	#$3600	#$3E00

Figure 3

PCLEAR	1	2	3	4	5	6	7	8
W/Out Disk	0C	12	18	1E	24	2A	30	36
With Disk	14	1A	20	26	2C	32	38	3E

```
LDB  #$( * )      * use number from table above
STB  $19
 - OR -
LDB  #(1 - 8)     PCLEAR 1 to PCLEAR 8
LDA  #6
MUL
ADDB $BC          #$E00 if disk, else #$600
STB  $19
```

Figure 4

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.

in the process, wipe out the buffer. To avoid this problem we must change the contents of $FF/100.

Graphic pages 1 to 4 are from $600 to $1DFF (with disk, from $E00 to $25FF), and we must put the edit buffer above graphics. Whatever number we put into Location $FF/100 must end in 00 and allow enough room for the text program before the ORG address. Let's use $2800 for the buffer address at $FF/100. This will allow locations $2800 to $3000 for the text and symbols and leave $3000 and up for the assembled program. This is done in the following manner:

1) insert *EDTASM+* cartridge and power up
2) press Z and ENTER to get into *ZBUG*
3) press W and ENTER to read two bytes at a time
4) type FF/ to look at $FF/100
5) type 2800 and press ENTER to change to #$2800
6) type GC006 and press ENTER to execute $C006 (keeps $FF/100 the same)

The assembler will now store the written program and symbols in a buffer starting at $2800. You can actually read the program in the "A" mode starting at $2A00. The "S" stack starts at the buffer location plus #$177, decreasing from there.

The next problem arises if you try to use '5' to '8' in the PAGE command at $9653. Since the computer is set only for PCLEAR 4, any higher number will give you a Function Call error message. To avoid this, we must PCLEAR some more pages. The BASIC PCLEAR command affects the addresses shown in Figure 3.

These numbers change as your BASIC program increases and becomes more complicated, but their initial value is assigned by the PCLEAR you select. So, in machine language, any PCLEAR must be put at least into Location $19. Either the program or table shown in Figure 4 can be used to get the PCLEAR HEX value you desire.

The second method, while a few bytes longer, is preferable because it will recognize whether or not disk is being used.

If you want to use PCLEAR 8, the *EDTASM+* buffer must begin at $3600 ($3E00 with disk) or higher. The program must execute at an address even higher — generally the length of your text program plus an additional 200 bytes. The *EDTASM+* will give you a Bad Memory error message if you try to write your program over the execution address. If it does, increase the ORG location until it is above the text portion of your program.

The program for this article is in machine language only. It can be executed entirely from *ZBUG* and will break when you hit any key without losing the program. Initially, Location $FF/100 is set to #$3E00 to get above disk graphics; the program will ORG at $4382 to leave room for the editor buffer and symbol table. Once you've typed in the program, switch to *ZBUG*. In the "A" mode you can follow the program and symbols from $4000 to $4381. That is why the program must ORG at least at $4382.

Once the program has been checked for errors, you can examine the buffer locations in the "W" mode for the following information.

Location	
buffer start +02	program end +1
+0E	symbols start at
+10	symbols end at
+40	"S" stack location
+9F	text start
+A5	text end
+A9	text start
+AB	text end

The location in Buffer +10 plus 1 will give you the lowest ORG location that will not conflict with the buffer.

Using the EQU command makes typing in routine addresses unnecessary and makes the program easier to follow. Eight pages are cleared; the program is set for PMODE 3,1; PCLS. The screen is filled, then displayed with SCREEN 1,1. The program switches to PMODE 3,5 and then fills and displays the screen. The action keeps alternating until you hit any key — JSR ($A000). When you do, the computer is reset for text screen and the program ends. In *ZBUG*, type FINISH = to see that the program ends at $43DA. Type FINISH - PCLEAR +1 = to get the length of the program, which is #$59 bytes.

A good technique to prevent the slight flicker on the graphics screen when alternating pages is to fill the screen first, then display it with the SCREEN command. This also gives the best animation effect. Since the program starts with PCLEAR, type GPCLEAR or G4382 to execute it. The END must be followed with PCLEAR.

You are not limited, by the way, to eight graphic pages of #$600 bytes each as long as you have enough memory to go higher without running into the edit buffer or the assembled program. The PCLEAR table continues in Figure 5. Generally, though, PCLEAR 12 is the highest you will use. The buffer must start at least at $4E00 ($5600 with disk) and execute even higher (text programs plus symbol length plus #$200).

(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.) □

PCLEAR	9	10	11	12	13	14	15	16	17	18
W/Out Disk	3C	42	48	4E	54	5A	60	66	6C	72
With Disk	44	4A	50	56	5C	62	68	6E	74	--

Figure 5

The listing: PAGER

```
                                                           EDTASM+/Ø1.ØØ.ØØ     PAGE 1

                        ØØØ5Ø *  $FF/1ØØ=#$3EØØ
4382                    ØØ1ØØ         ORG     $4382
          9628          ØØ11Ø PMODE   EQU     $9628
          9653          ØØ12Ø PAGE    EQU     $9653
          9542          ØØ13Ø PCLS    EQU     $9542
          95AA          ØØ14Ø SCREEN  EQU     $95AA
          9682          ØØ15Ø CSET    EQU     $9682
4382 C6   Ø8            ØØ16Ø PCLEAR  LDB     #8        FOR 8 PAGES
4384 86   Ø6            ØØ17Ø         LDA     #6        #$6ØØ BYTES PER PAGE
4386 3D                 ØØ18Ø         MUL
4387 DB   BC            ØØ19Ø         ADDB    $BC       WITH OR WITHOUT DISK?
4389 D7   19            ØØ2ØØ         STB     $19       WHERE BASIC WOULD START
438B C6   Ø3            ØØ21Ø         LDB     #3        PMODE 3
438D BD   9628          ØØ22Ø         JSR     PMODE
439Ø C6   Ø1            ØØ23Ø PAGE1   LDB     #1        PAGE 1
4392 BD   9653          ØØ24Ø         JSR     PAGE
4395 BD   9542          ØØ25Ø         JSR     PCLS
4398 9E   BA            ØØ26Ø         LDX     $BA       START OF GRAPHICS ON PAGE 1
439A CC   ØØØØ          ØØ27Ø         LDD     #Ø
439D ED   81            ØØ28Ø LOOP1   STD     ,X++
439F C3   ØØØ3          ØØ29Ø         ADDD    #3
43A2 9C   B7            ØØ3ØØ         CMPX    $B7       END OF GRAPHICS ON PAGE 1
43A4 25   F7            ØØ31Ø         BLO     LOOP1
43A6 C6   Ø1            ØØ32Ø         LDB     #1
43A8 BD   95AA          ØØ33Ø         JSR     SCREEN    DISPLAY THE SCREEN
43AB C6   Ø1            ØØ34Ø         LDB     #1
43AD BD   9682          ØØ35Ø         JSR     CSET      COLOR SET 1
43BØ C6   Ø5            ØØ36Ø         LDB     #5        PAGE 5
43B2 BD   9653          ØØ37Ø         JSR     PAGE
43B5 BD   9542          ØØ38Ø         JSR     PCLS
43B8 9E   BA            ØØ39Ø         LDX     $BA       START OF GRAPHICS ON PAGE 5
43BA CC   ØØØØ          ØØ4ØØ         LDD     #Ø
43BD ED   81            ØØ41Ø LOOP5   STD     ,X++
43BF 83   ØØØ3          ØØ42Ø         SUBD    #3
43C2 9C   B7            ØØ43Ø         CMPX    $B7
43C4 25   F7            ØØ44Ø         BLO     LOOP5
43C6 C6   Ø1            ØØ45Ø         LDB     #1
43C8 BD   95AA          ØØ46Ø         JSR     SCREEN
43CB C6   Ø1            ØØ47Ø         LDB     #1
43CD BD   9682          ØØ48Ø         JSR     CSET
43DØ AD   9F AØØØ       ØØ49Ø DONE    JSR     [$AØØØ]
43D4 27   BA            ØØ5ØØ         BEQ     PAGE1
43D6 5F                 ØØ51Ø         CLRB
43D7 BD   95AA          ØØ52Ø         JSR     SCREEN
43DA 3F                 ØØ53Ø FINISH  SWI               RTS IF IN BASIC
          4382          ØØ54Ø         END     PCLEAR

ØØØØØ TOTAL ERRORS
```