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

Machine Language Made BASIC:
Part VII: Around in Circles

By William P. Nee

The circle routine in ROM is certainly one of the most difficult to understand and use. The routine keeps alternating between its BASIC check for syntax, format, etc., and short subroutines required as set-up for the main routine. Since the BASIC checks would interfere with a machine language program, we must bypass them and implement the subroutine on our own. The following locations are used in the circle routine:

Location	
$B5	color
** $CB/CC	horizontal center of circle (X)
** $CD/CE	vertical center of circle (Y)
** $CF/D0	radius
** $D1/D2	height to width ratio
** $D3/D4	maximum horizontal coordinate
** $D5/D6	maximum vertical coordinate
$D8	circle/line flag 1: = circle, 0 = line
$D9/DA	arc end

The locations marked with two asterisks (**) must be scaled for locations, or dimensions, in any PMODE other than PMODE 4. The CIRCLE routine may be executed from three different addresses, depending on how much of the CIRCLE command you use.

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.

The first step is to load the maximum coordinates into $D3/D4 and $D5/D6 by using the subroutine at Address $9522. This routine not only loads the correct numbers according to the PMODE, it also scales them. Next, enter the coordinates of the circle's center. This is usually done by loading Register D with the X,Y locations (i.e., #$8060 would be the center of the screen), then storing Register A in $0CC and Register B in $CE. We must show Location $CC as $0CC because *EDTASM+* requires a zero in front of any number that could also be a register, such as A, B, CC, or D.

These locations are scaled by loading Stack U with #$CB (start of X location) followed by JSR $9320. Next, load Register A (LDA) with the radius, and store it in Location $D0. Scale it by loading Stack U with #$CF followed by JSR $9320. Load Register B with the desired color byte and store it in $B5.

The height-to-width subroutine at $9EC9 converts the desired height/width ratio to a two-byte number and stores it in Location $D1/D2. The following chart identifies the number to be placed in Location $D1/D2 for each height-to-width ratio and possible PMODE screen. The numbers for PMODE 3 and 2 are twice those of PMODE 4, 1, and 0.

Height-to-Width Ratio	PMODE 4, 1 or 0	PMODE 3 or 2
.1	#$0019	#$0032
.2	#$0033	#$0066
.3	#$004C	#$0098
.4	#$0066	#$00CC
.5	#$0080	#$0100
.6	#$0099	#$0132
.7	#$00B3	#$0166
.8	#$00CC	#$0198
.9	#$00EC	#$01D8
1.0	#$0100	#$0200
1.5	#$0180	#$0300
2.0	#$0200	#$0400
2.5	#$0280	#$0500

If you want your program to run in any PMODE, include the subroutine, which checks the PMODE and doubles the height-to-width ratio when necessary:

```
        LDX     #$ **     ** use two-byte
                          number from chart
        LDA     $B6       get PMODE
        BITA    #2        register A AND 2
        BEQ     NEXT      PMODE 4, 1,
                          or 0 will equal 0
        TFR     X,D       height-to-width
                          ratio to Register D
        LEAX    D,X       double it
NEXT    STX     $D1
```

Store the number 1 in Location $D8 to indicate that no lines should be drawn. Try clearing this location in a program and see what happens.

The arc subroutine at $9FED converts an arc decimal to a two-byte number stored in Location $D9/DA.

Many of these arcs and their two-byte equivalents are in the following chart:

Arc in Degrees	Arc in Degrees
0 = #$0600	210 = #$0205
30 = #$0605	225 = #$0300
45 = #$0700	240 = #$0302
60 = #$0702	270 = #$0400
90 = #$0000	300 = #$0405
120 = #$0005	315 = #$0500
135 = #$0100	330 = #$0502
150 = #$0102	360 = #$0600
180 = #$0200	

Load Register D with the desired arc end two-byte equivalent, and store it in Location $D9/DA. Then load Register D with the arc-start two-byte equivalent and JSR $9EFD. Be sure to save whatever is in registers A or B prior to executing this routine if you will need them again.

Listing 1 begins by drawing a circle in the center of the screen. It continues drawing circles — each two steps larger than the previous one — until the radius reaches 90. Instead of using the standard start-up assigned by the computer, COLOR 3,0, this program uses COLOR 0,3, which sets a buff background and draws black circles.

If the height-to-width ratio is 1 and the routine draws a full circle with no arcs, you can stop after scaling the radius and JSR $9EC2. Again, be sure that you have already saved registers A and B. This routine also checks Location $C2 for a PSET or PRESET, stores a '1' in Location $D8 (which indicates that no lines will be drawn), and assigns the foreground color to Location $B5.

If you want to specify a height-to-width ratio other than 1, load that ratio's two-byte equivalent into Register X, the desired color into $B5, and then load JSR $9EDF. Using this address, the height-to-width ratio will automatically be scaled (doubled) for PMODE 3 or 2. The routine also checks Location $C2 for a PSET or PRESET and stores a '1' in Location $D8.

Assigning arc starts and arc ends will require that you use the height-to-width ratio chart described earlier in this article. Store the correct two-byte height-to-width ratio (according to the PMODE) in $D1/D2; store the two-byte arc end in $D9/DA; load Register D with the two-byte arc start, then with JSR $9EFD.

More than one arc can be drawn by repeating the arc portion of the program as often as desired. Color changes can be made by putting the desired color into Location $B5 prior to executing the routine. Listing 2 is a program that draws different colored arcs of 60 degrees at increasing radii.

Since the first program is in PMODE 4, I didn't need to scale any of the numbers, but I did so to show how it is done. It is good practice to always scale since you may want to change the PMODE later.

The routines for drawing circles or arcs are found in Figure 1.

Before assembling either program, set Location $FF/100 to #$2000 and execute GC006. If you run the assembled programs from BASIC, clear enough memory by entering CLEAR 200, &H3000-1, and change the SWI in Line 670 to RTS. □

```
JSR $9EC2            JSR $9EDF        JSR $9EFD
SET MAX X,Y          PLUS -           PLUS -
SET CENTER/SCALE     SET COLOR        STORE "1" IN $D8
SET RADIUS/SCALE     SET H/W RATIO    SET ARC END
                                      SET ARC START
```

Figure 1: Routines for Drawing Circles or Arcs

Listing 1: CIRCLES

```
                       ØØ1ØØ *$FF/1ØØ=#$2ØØØ
3ØØØ                   ØØ11Ø       ORG    $3ØØØ
3ØØØ C6    Ø4          ØØ12Ø START LDB    #4        PMODE 4
3ØØ2 BD    9628        ØØ13Ø       JSR    $9628
3ØØ5 C6    Ø1          ØØ14Ø       LDB    #1        PAGE 1
3ØØ7 BD    9653        ØØ15Ø       JSR    $9653
3ØØA ØF    B2          ØØ16Ø       CLR    $B2       CLEAR FOREGROUND
3ØØC C6    Ø3          ØØ17Ø       LDB    #3
3ØØE D7    B3          ØØ18Ø       STB    $B3       BACKGROUND COLOR
3Ø1Ø BD    9542        ØØ19Ø       JSR    $9542     COLOR Ø,3
3Ø13 C6    Ø1          ØØ2ØØ       LDB    #1        GRAPHICS SCREEN
3Ø15 BD    95AA        ØØ21Ø       JSR    $95AA
3Ø18 C6    Ø1          ØØ22Ø       LDB    #1        COLOR SET 1
3Ø1A BD    9682        ØØ23Ø       JSR    $9682
3Ø1D BD    9522        ØØ24Ø       JSR    $9522     SET MAXIMUM X,Y AND SCALE THEM
3Ø2Ø CC    8Ø6Ø        ØØ25Ø       LDD    #$8Ø6Ø    CENTER OF THE CIRCLE(128,96)
3Ø23 97    CC          ØØ26Ø       STA    $ØCC      X LOCATION
3Ø25 D7    CE          ØØ27Ø       STB    $CE       Y LOCATION
3Ø27 CE    ØØCB        ØØ28Ø       LDU    #$CB      START OF X LOCATION
3Ø2A BD    932Ø        ØØ29Ø       JSR    $932Ø     SCALE X,Y
3Ø2D C6    Ø2          ØØ3ØØ       LDB    #2        SIZE OF FIRST RADIUS
3Ø2F 34    Ø4          ØØ31Ø LOOP  PSHS   B         SAVE RADIUS
```

```
3Ø31 D7    DØ      ØØ32Ø         STB   $DØ      RADIUS LOCATION
3Ø33 CE    ØØCF    ØØ33Ø         LDU   #$CF     START OF RADIUS LOCATION
3Ø36 BD    932Ø    ØØ34Ø         JSR   $932Ø    SCALE RADIUS
3Ø39 BD    9EC2    ØØ35Ø CIRCLE  JSR   $9EC2    CIRCLE(X,Y),R
3Ø3C 35    Ø4      ØØ36Ø         PULS  B        GET THE RADIUS
3Ø3E CB    Ø2      ØØ37Ø         ADDB  #2       INCREASE IT
3Ø4Ø C1    5A      ØØ38Ø         CMPB  #9Ø      MAXIMUM RADIUS YET?
3Ø42 23    EB      ØØ39Ø         BLS   LOOP
3Ø44 BD    ADFB    ØØ4ØØ DONE    JSR   $ADFB    WAIT FOR INPUT
3Ø47 5F            ØØ41Ø         CLRB           TEXT SCREEN
3Ø48 BD    95AA    ØØ42Ø         JSR   $95AA
3Ø4B 3F            ØØ43Ø         SWI            RTS IF IN BASIC
           3ØØØ    ØØ44Ø         END   START

ØØØØØ TOTAL ERRORS
```

Listing 2: ARCS

```
                   ØØ1ØØ * $FF/1ØØ=#$2ØØØ
3ØØØ               ØØ11Ø         ORG   $3ØØØ
3ØØØ C6    Ø3      ØØ12Ø START   LDB   #3       PMODE 3
3ØØ2 BD    9628    ØØ13Ø         JSR   $9628
3ØØ5 C6    Ø1      ØØ14Ø         LDB   #1       PAGE 1
3ØØ7 BD    9653    ØØ15Ø         JSR   $9653
3ØØA BD    9542    ØØ16Ø         JSR   $9542    PCLS
3ØØD C6    Ø1      ØØ17Ø         LDB   #1       GRAPHICS SCREEN
3ØØF BD    95AA    ØØ18Ø         JSR   $95AA
3Ø12 C6    Ø1      ØØ19Ø         LDB   #1       COLOR SET 1
3Ø14 BD    9682    ØØ2ØØ         JSR   $9682
3Ø17 BD    9522    ØØ21Ø         JSR   $9522    SET MAXIMUM X,Y AND SCALE THEM
3Ø1A CC    8Ø6Ø    ØØ22Ø         LDD   #$8Ø6Ø   CENTER OF CIRCLE(128,96)
3Ø1D 97    CC      ØØ23Ø         STA   $ØCC     X LOCATION
3Ø1F D7    CE      ØØ24Ø         STB   $CE      Y LOCATION
3Ø21 CE    ØØCB    ØØ25Ø         LDU   #$CB     START OF X LOCATION
3Ø24 BD    932Ø    ØØ26Ø         JSR   $932Ø    SCALE X,Y
3Ø27 C6    Ø2      ØØ27Ø         LDB   #2       SIZE OF FIRST RADIUS
3Ø29 34    Ø4      ØØ28Ø LOOP    PSHS  B        SAVE THE RADIUS
3Ø2B D7    DØ      ØØ29Ø         STB   $DØ      RADIUS LOCATION
3Ø2D CE    ØØCF    ØØ3ØØ         LDU   #$CF     START OF RADIUS LOCATION
3Ø3Ø BD    932Ø    ØØ31Ø         JSR   $932Ø    SCALE THE RADIUS
3Ø33 8E    Ø1ØØ    ØØ32Ø         LDX   #$Ø1ØØ   HEIGHT TO WIDTH RATIO = 1
3Ø36 96    B6      ØØ33Ø         LDA   $B6      PMODE
3Ø38 85    Ø2      ØØ34Ø         BITA  #2       PMODE 4,1, OR Ø WILL = Ø
3Ø3A 27    Ø4      ØØ35Ø         BEQ   NEXT
3Ø3C 1F    1Ø      ØØ36Ø         TFR   X,D      REGISTER X TO REGISTER D
3Ø3E 3Ø    8B      ØØ37Ø         LEAX  D,X      ADD REGISTER D TO REGISTER X
3Ø4Ø 9F    D1      ØØ38Ø NEXT    STX   $D1      HEIGHT TO WIDTH RATIO LOCATION
3Ø42 CC    Ø155    ØØ39Ø         LDD   #$Ø155
3Ø45 97    D8      ØØ4ØØ         STA   $D8      NO LINES, JUST CIRCLES
3Ø47 D7    B5      ØØ41Ø         STB   $B5      COLOR
3Ø49 CC    Ø7Ø2    ØØ42Ø         LDD   #$Ø7Ø2   END OF ARC 1 - 6Ø DEGREES
3Ø4C DD    D9      ØØ43Ø         STD   $D9      END OF ARC LOCATION
3Ø4E CC    Ø6ØØ    ØØ44Ø         LDD   #$Ø6ØØ   START OF ARC 1 - Ø DEGREES
```

```
3Ø51 BD     9EFD      ØØ45Ø         JSR     $9EFD     DRAW THE ARC
3Ø54 CC     Ø1AA      ØØ46Ø         LDD     #$Ø1AA
3Ø57 97     D8        ØØ47Ø         STA     $D8       NO LINES
3Ø59 D7     B5        ØØ48Ø         STB     $B5       COLOR
3Ø5B CC     Ø2ØØ      ØØ49Ø         LDD     #$Ø2ØØ    END OF ARC 2 - 18Ø DEGREES
3Ø5E DD     D9        ØØ5ØØ         STD     $D9       END OF ARC LOCATION
3Ø6Ø CC     ØØØ5      ØØ51Ø         LDD     #$ØØØ5    START OF ARC 2 - 12Ø DEGREES
3Ø63 BD     9EFD      ØØ52Ø         JSR     $9EFD     DRAW THE ARC
3Ø66 CC     Ø1FF      ØØ53Ø         LDD     #$Ø1FF
3Ø69 97     D8        ØØ54Ø         STA     $D8       NO LINES
3Ø6B D7     B5        ØØ55Ø         STB     $B5       COLOR
3Ø6D CC     Ø4Ø5      ØØ56Ø         LDD     #$Ø4Ø5    END OF ARC 3 - 3ØØ DEGREES
3Ø7Ø DD     D9        ØØ57Ø         STD     $D9       START OF ARC LOCATION
3Ø72 CC     Ø3Ø2      ØØ58Ø         LDD     #$Ø3Ø2    START OF ARC 3 - 24Ø DEGREES
3Ø75 BD     9EFD      ØØ59Ø         JSR     $9EFD     DRAW THE ARC
3Ø78 35     Ø4        ØØ6ØØ         PULS    B         GET THE RADIUS
3Ø7A CB     Ø2        ØØ61Ø         ADDB    #2        INCREASE IT
3Ø7C C1     5A        ØØ62Ø         CMPB    #9Ø       MAXIMUM RADIUS YET?
3Ø7E 23     A9        ØØ63Ø         BLS     LOOP
3Ø8Ø BD     ADFB      ØØ64Ø DONE    JSR     $ADFB     WAIT FOR INPUT
3Ø83 5F               ØØ65Ø         CLRB              TEXT SCREEN
3Ø84 BD     95AA      ØØ66Ø         JSR     $95AA
3Ø87 3F               ØØ67Ø         SWI               RTS IF IN BASIC
            3ØØØ      ØØ68Ø         END     START

ØØØØØ TOTAL ERRORS
```