I present you super fast routine for drawing circles in graphics mode 8 and 8 + 16.

**Moj Mikro** magazine article listings are available as zipped file in download section. It contains ATR disk image file with Atari DOS system for using with disk based, real Atari machine or any Atari 8-bit emulator. To load Atari BASIC programs properly, Atari must be powered with Atari BASIC enabled, or you can try them with any compatible BASIC.

You can try, use and modify the programs and routines for whatever purpose you want.

## Super Fast Circle Routine#

- Magazine: Moj Mikro, 1989/3
- Author : Zlatko Bleha
- Page : 27 - 31
- Article link
- Atari BASIC listing on disk (tokenized): M8903284.BAS
- Atari BASIC listing (listed): M8903284.LST
- Assembly language listing: CIRCLE.LST

This routine uses very fast algorithm for drawing circles. It works in graphics mode 8. To see it in action, try the following demonstration program, which includes circle routine in DATA tables for using it from Atari BASIC. Also included is assembly language source code for anybody to examine inner workings of the algorithm. It is written in ATMAS 2.

## Basic Listing: M8903284.LST#

```plaintext
1 REM ************************************
2 REM PROGRAM : SUPER FAST CIRCLE ROUTINE
3 REM AUTHOR   : ZLATKO BLEHA
4 REM PUBLISHER: MOJ MIKRO MAGAZINE
5 REM ISSUE NO.: 1989, NO.3, PAGE 28
6 REM ************************************
7 REM
```
10 FOR A=30000 TO 30288
20 READ B:C=C+B:POKE A,B
30 NEXT A
40 IF C<>28218 THEN ? "*** DATA ERROR ***":END
50 DATA 120,104,104,104,133,58,104,133,51,133
60 DATA 60,104,133,50,133,59,104,104,168,133
70 DATA 54,104,104,208,5,152,88,76,242,117
80 DATA 133,56,56,233,1,133,55,160,0,132
90 DATA 57,165,59,24,101,57,133,50,152,101
100 DATA 60,133,51,165,54,101,56,133,61,32
110 DATA 242,117,165,54,56,229,56,133,62,32
120 DATA 242,117,165,59,56,229,57,133,50,165
130 DATA 60,233,0,133,51,165,62,32,242,117
140 DATA 165,61,32,242,117,165,59,24,101,56
150 DATA 133,50,152,101,60,133,51,165,54,101
160 DATA 57,133,63,32,242,117,165,54,56,229
170 DATA 57,133,64,32,242,117,165,59,56,229
180 DATA 56,133,50,165,60,233,0,133,51,165
190 DATA 64,32,242,117,165,63,32,242,117,230
200 DATA 57,165,55,56,229,57,229,57,16,7
210 DATA 198,56,24,101,56,101,56,133,55,165
220 DATA 56,197,57,48,3,76,89,117,96,104
230 DATA 104,104,133,58,104,133,51,104,133,50
240 DATA 104,104,160,0,132,49,133,48,10,38
250 DATA 49,10,38,49,24,101,48,170,152,101
260 DATA 49,133,49,138,10,38,49,10,38,49
270 DATA 10,38,49,24,101,88,133,48,165,49
280 DATA 101,89,133,49,166,51,134,53,165,50
290 DATA 170,70,53,106,70,53,106,70,53,106
300 DATA 24,101,48,133,48,165,53,101,49,133
310 DATA 49,138,41,7,170,152,56,106,202,16
320 DATA 252,166,58,240,6,17,48,145,48,88
330 DATA 96,73,255,49,48,145,48,88,96
500 REM ****************************************
501 REM * DEMONSTRATION PROGRAM *
502 REM ****************************************
503 REM
510 GRAPHICS 8:SETCOLOR 2,0,0:COLOR 3
520 FOR A=2 TO 90
530 Q=USR(30000,1,A,A,A)
540 Q=USR(30000,1,319-A,159-A,A)
550 NEXT A
560 ? "PRESS ANY KEY TO CONTINUE"
570 IF PEEK(555)=0 THEN 570
590 Q=USR(30000,1,319-A,159-A,A)
600 Q=USR(30000,1,A,159-A,A)
610 NEXT A
620 ? "PRESS ANY KEY TO CONTINUE"
630 IF PEEK(555)=0 THEN 630
640 GRAPHICS 8:SETCOLOR 2,0,0:COLOR 3
650 FOR A=2 TO 79 STEP 4
660 Q=USR(30000,1,100,A,A)
670 NEXT A
680 FOR A=82 TO 2 STEP -4
690 Q=USR(30000,1,A+100,A,A)
700 NEXT A
710 ? "PRESS ANY KEY TO START AGAIN"
720 IF PEEK(555)=0 THEN 720
ATAMAS 2 Assembler Listing CIRCLE.LST

* FAST CIRCLE DRAWING ROUTINE
* Author   : Zlatko Bleha
* Publisher: Moj Mikro magazine
* Issue No.: 1989, No.3, Page 27
* System variables on Page Zero

LO EQU $30
HI EQU $31
XLO EQU $32
XHI EQU $33
XLOH EQU $34
XHIH EQU $35
Y EQU $36
A EQU $37
B EQU $38
C EQU $39
POK EQU $3A
X1 EQU $3B
X2 EQU $3C
YVIB EQU $3D
YMAB EQU $3E
YVIC EQU $3F
YMAC EQU $40

ORG $7530
SEI
PLA
PLA
PLA
STA POK
PLA
STA XHI
STA X2
PLA
STA XLO
STA X1
PLA
PLA
TAY
STA Y
PLA
PLA
BNE NEXT
TYA
CLI
JMP PLOT
NEXT STA B
SEC
SBC #$1
STA A
LDY #$0
STY C

*  -----------------------------
*  PLOT X+C,Y+B
*  -----------------------------

LOOP LDA X1
    CLC
    ADC C
    STA XLO
    TYA
    ADC X2
    STA XHI
    LDA Y
    ADC B
    STA YVIB
    JSR PLOT

*  -----------------------------
*  PLOT X+C,Y-B
*  -----------------------------

    LDA Y
    SEC
    SBC B
    STA YMAB
    JSR PLOT

*  -----------------------------
*  PLOT X-C,Y-B
*  -----------------------------

    LDA X1
    SEC
    SBC C
    STA XLO
    LDA X2
    SBC #$0
    STA XHI
    LDA YMAB
    JSR PLOT

*  -----------------------------
*  PLOT X-C,Y+B
*  -----------------------------

    LDA YVIB
    JSR PLOT

*  -----------------------------
*  PLOT X+B,Y+C
LDA X1
CLC
ADC B
STA XLO
TYA
ADC X2
STA XHI
LDA Y
ADC C
STA YVIC
JSR PLOT

PLOT X+B,Y-C

LDA Y
SEC
SBC C
STA YMAC
JSR PLOT

PLOT X-B,Y-C

LDA X1
SEC
SBC B
STA XLO
LDA X2
SBC #$0
STA XHI
LDA YMAC
JSR PLOT

PLOT X-B,Y+C

LDA YVIC
JSR PLOT

INC C
LDA A
SEC
SBC C
SBC C
BPL JUMP
DEC B
CLC
ADC B
ADC B
JUMP STA A
LDA B
CMP C
BMI END
JMP LOOP
END RTS
*
* -----------------------------
* PLOT routine
* -----------------------------
*
* Taking parameters from stack in case
* PLOT routine is called directly from
* BASIC
* -----------------------------
*
PLA
PLA
PLA
STA POK
PLA
STA XHI
PLA
STA XLO
PLA
PLA
LDY #$0
*
* -----------------------------
* Beginning of the PLOT routine
* -----------------------------
*
* Calculating A*40
* -----------------------------
*
PLOT STY HI
STA LO
ASL
ROL HI
ASL
ROL HI
CLC
ADC LO
TAX
TYA
ADC HI
STA HI
TXA
ASL
ROL HI
ASL
ROL HI
ASL
ROL HI
*
* -----------------------------
* A*40+VIDEO
* -----------------------------
*
CLC
ADC $58
STA LO
LDA HI
ADC $59
STA HI

* -----------------------------
* X/8  -----------------------------
*
LDX XHI
STX XHIH
LDA XLO
TAX
LSR XHIH
ROR
LSR XHIH
ROR
LSR XHIH
ROR

* -----------------------------
* A*40+VIDEO+X/8  -----------------------------
*
CLC
ADC LO
STA LO
LDA XHIH
ADC HI
STA HI

* -----------------------------
* Calculating a location of  
* destination bit  
* -----------------------------
*
TXA
AND #$7
TAX
TYA
SEC
ROT
ROR
DEX
BPL ROT

* -----------------------------
* PLOT or UNPLOT?  
* -----------------------------
*
LDX POK
BEQ UNPLOT

* -----------------------------
* PLOT  
* -----------------------------
*
ORA (LO),Y
STA (LO),Y
CLI
RTS
Description

Algorithm is based on one from Croatian computer magazine Racunari (number 31). There is a little faster way of doing this, but data must be provided in tables for sine and cosine. But this alternative takes more RAM.

The routine consists of two parts: main program for calculating X and Y coordinates and second part with PLOT routine for drawing (or erasing) pixels on X and Y coordinates. Calculation of coordinates is done for one eighth of the circle. Other seven parts are simply mirrored from original mentioned pattern. We take advantage of this fact, because circle is symmetrically shaped. The routine uses Page Zero locations for variables for even faster execution. There are 17 of them.

Usage of the routine

A=USR(30000,POK,X,Y,R)

<table>
<thead>
<tr>
<th>30000</th>
<th>Starting address of the calling routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>POK</td>
<td>1 (draw the circle), 0 (erase the circle)</td>
</tr>
<tr>
<td>X</td>
<td>X coordinate of the circle</td>
</tr>
<tr>
<td>Y</td>
<td>Y coordinate of the circle</td>
</tr>
<tr>
<td>R</td>
<td>Radius of the circle</td>
</tr>
</tbody>
</table>

The routine is not relocatable. If you want to relocate it anyway, use assembler of your choice and assemble it to another location or amend the DATA tables in Atari BASIC program demonstration.

Maximum value for radius is 127, which is more than enough, because the circle with such radius will not be seen on the screen in whole anyway. If you choose wrong parameter values and consequently break the borders of the screen, don't worry, circle will be drawn anyway, but with its parts in different places. Experiment with the values for best results and your own fun.

The included assembly language listing shows what is necessary for implementing circle algorithm, calculating coordinates and drawing pixels and circles on the screen.

Fast circle drawing in pure Atari BASIC

- Magazine: Moj Mikro, 1989/3
- Author: Zlatko Bleha
- Page: 27 - 31
- Atari BASIC listing on disk (tokenized): M8903282.BAS
- Atari BASIC listing (listed): M8903282.LST
Next example is demonstration of implementing mentioned circle algorithm in pure Atari BASIC. This program shows how much faster it is compared to classic program using sine and cosine functions from Atari BASIC (shown in last example).

Basic Listing M8903282.LST

1 REM ****************************
2 REM PROGRAM : FAST CIRCLE DRAWING
3 REM AUTHOR : ZLATKO BLEHA
4 REM PUBLISHER: MOJ MIKRO MAGAZINE
5 REM ISSUE NO.: 1989, NO.3, PAGE 29
6 REM ****************************
7 REM
10 GRAPHICS 8:SETCOLOR 2,0,0:COLOR 3
20 ? "ENTER X, Y AND R"
30 INPUT X,Y,R
40 IF R=0 THEN PLOT X,Y:END
50 B=R:C=0:A=R-1
60 PLOT X+C,Y+B
70 PLOT X+C,Y-B
80 PLOT X-C,Y-B
90 PLOT X-C,Y+B
100 PLOT X+B,Y+C
110 PLOT X+B,Y-C
120 PLOT X-B,Y-C
130 PLOT X-B,Y+C
140 C=C+1
150 A=A+1-C-C
160 IF A>=0 THEN 190
170 B=B-1
180 A=A+B+B
190 IF B>=C THEN 60

Use some valid values for coordinates and radius, for example:

X=40, Y=40, R=30
X=130, Y=90, R=60

Slow circle drawing in Atari BASIC

- Magazine: Moj Mikro, 1989/3
- Author : Zlatko Bleha
- Page : 27 - 31
- Atari BASIC listing on disk (tokenized): M8903281.BAS
- Atari BASIC listing (listed): M8903281.LST

This is classic example for drawing circles from Atari BASIC using sine and cosine functions. Unfortunately, this is very slow way of doing it and not recommended. Just use routine shown above and everybody will be happy.
Basic Listing M8903281.LST#

1 REM  ****************************************
2 REM PROGRAM  : SLOW CIRCLE DRAWING
3 REM AUTHOR   : ZLATKO BLEHA
4 REM PUBLISHER: MOJ MIKRO MAGAZINE
5 REM ISSUE NO.: 1989, NO.3, PAGE 29
6 REM  ****************************************
7 REM
10 GRAPHICS 8:SETCOLOR 2,0,0:COLOR 3
20 FOR A=0 TO 6.28 STEP 0.02
30 X=SIN(A)*50+150
40 Y=COS(A)*50+80
50 PLOT X,Y
60 NEXT A

Conclusion#

Returning back to first program with the fastest way of drawing circles... There is one more thing to note. In case you want to use PLOT subroutine, which is part of the main circle routine, then read following explanation.

PLOT routine is written so it can be used easily from Atari BASIC program independently from main circle routine, by using like this:

A=USR(30179,POK,X,Y)

<table>
<thead>
<tr>
<th>POK</th>
<th>1 (drawing a pixel), 0 (erasing a pixel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X coordinate of the pixel</td>
</tr>
<tr>
<td>Y</td>
<td>Y coordinate of the pixel</td>
</tr>
</tbody>
</table>

The routine alone is not any faster than normal PLOT command from Atari BASIC, because USR command takes approximately 75% of whole execution. But, used as part of the main circle routine it does not matter anymore, because it is integrated in one larger entity. There the execution is very fast, with no overhead. PLOT routine is here for you to examine anyway. You never know if you will maybe need it in the future.

Greetings, Gury