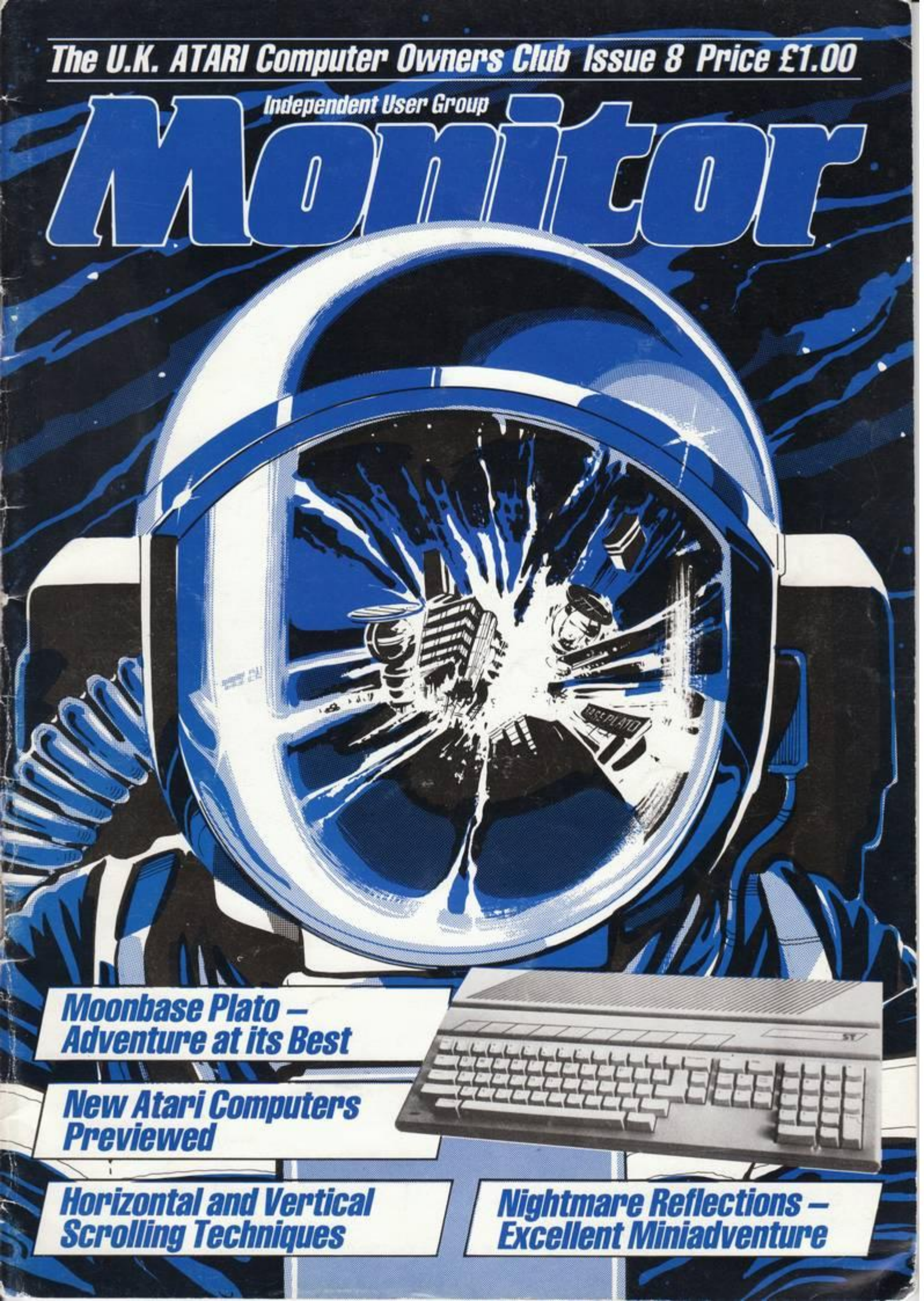
The U.K. ATARI Computer Owners Club Issue 8 Price £1.00

Independent User Group

Monitor



Moonbase Plato – Adventure at its Best

New Atari Computers Previewed

Horizontal and Vertical Scrolling Techniques

Nightmare Reflections – Excellent Miniadventure

QUICKPLOT

A FAST GRAPHICS 8 PLOT-DRAWTO HANDLER

BY PETER BLACKMORE

So far, there have been quite a number of Graphics 8 utilities presented in the magazine. These include an 80 column screen, text on Graphics 8, a data compressor, and a routine to print out the screen. With all these improvements, it's about time the Plot-Drawto function was brought up to par!

The Atari operating system does not use a particularly fast method to draw lines, a far better one is Bresenham's algorithm, devised in 1965. The way this algorithm works is best illustrated by example.

Suppose we have just Plotted (0,0) and we wish to Drawto (300,150). The Drawto function must calculate all the points between the two limits which must be filled in. It does this in the following manner.

The change in X and the change in Y, deltax and deltay respectively, are calculated. In this case they are 300 and 150. Since deltax > deltay, each time the X value of the point currently being plotted is altered, the Y co-ordinate may or may not change. In this case Y changes once for every two times X changes, see Diagram 1.

Bresenham's method keeps an error term to which is added the slope of the line being drawn (deltay/deltax) on each iteration. The sign of this error term then indicates whether the drawn line is above or below the ideal line, and hence whether or not the Y value should be changed.

The program listed below is an implementation of this algorithm, and used from Basic will draw lines in half the time that the normal Plot-Drawto takes. Listing 1 is for use from Basic and includes a number of demos. The Plot-Drawto is accessed by:–

A=USR(P,X,Y) for PLOT X,Y
A=USR(D,X,Y) for DRAWTO X,Y

P and D are set to 1536 and ADR(DRAW$) respectively. Remember to save a version out before running the program, and also that this routine will not detect errors such as PLOT 700,400, so make sure that you pass it the correct values or suffer the consequences!

Listing 2 is the assembly code for the Plot-Drawto function for all of you that want to speed it up even more, there are ways of doing this! If you know of a different method why not send it in!!

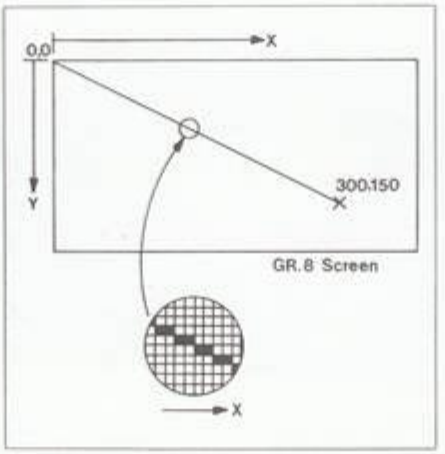


Diagram 1.

Listing 1.

```
1 DIM DRAW$(392)
2 FOR A=1 TO 392:READ OP:DRAW$(A,A)=CH
R$(OP):NEXT A
3 FOR A=1536 TO 1694:READ OP:POKE A,OP
:NEXT A
5 P=1539:D=ADR(DRAW$):DEL=300
10 GRAPHICS 24:POKE 709,11:POKE 710,0:
COLOR 1
20 FOR A=0 TO 319 STEP 2:X=USR(P,159,0
):X=USR(D,A,191):NEXT A
30 GOSUB DEL
40 FOR A=0 TO 319 STEP 3:X=USR(P,159,1
91):X=USR(D,A,0):NEXT A
50 GOSUB DEL
60 FOR A=0 TO 191 STEP 2:X=USR(P,0,95)
:X=USR(D,319,A):NEXT A
70 GOSUB DEL
80 FOR A=0 TO 191 STEP 3:X=USR(P,319,9
5):X=USR(D,0,A):NEXT A
90 GOSUB DEL
100 PLOT 8,8:DRAWTO 311,8:DRAWTO 311,1
83:DRAWTO 8,183:DRAWTO 8,8
110 FOR A=95 TO 10 STEP -2:X=USR(P,159
,95):X=USR(D,10,A):NEXT A
120 FOR A=10 TO 309 STEP 3:X=USR(P,159
,95):X=USR(D,A,10):NEXT A
130 FOR A=10 TO 181 STEP 2:X=USR(P,159
,95):X=USR(D,309,A):NEXT A
140 FOR A=309 TO 10 STEP -3:X=USR(P,15
9,95):X=USR(D,A,181):NEXT A
150 FOR A=181 TO 96 STEP -2:X=USR(P,15
9,95):X=USR(D,10,A):NEXT A
160 GOSUB DEL
170 FOR A=90 TO 158:X=USR(P,A,A):X=USR
(D,319-A,A):X=USR(D,319-A,191-A):X=USR
(D,A,191-A):X=USR(D,A,A):NEXT A
180 GOSUB DEL
190 FOR A=0 TO 95:X=USR(P,A,A):X=USR(D
,319-A,A):X=USR(D,319-A,191-A):X=USR(D
,A,191-A):X=USR(D,A,A):NEXT A
200 GOSUB DEL
210 FOR A=94 TO 0 STEP -3:X=USR(P,A,A)
:X=USR(D,319-A,A):X=USR(D,319-A,191-A)
:X=USR(D,A,191-A):X=USR(D,A,A):NEXT A
220 GOSUB DEL
299 END
300 FOR A=1 TO 500:NEXT A:? #6;CHR$(12
5);:RETURN
999 END
1000 DATA 104,201,2,208,118,104,141,16
8,6,104,141,167,6,104,208,107,104,141,
169,6,216,169,64,141,165,6
1010 DATA 141,166,6,169,0,141,163,6,14
1,164,6,173,167,6,56,229,85,141,170,6,
173,168,6,229,86,16
1020 DATA 24,14,165,6,73,255,141,171,6
,173,170,6,73,255,24,105,1,141,170,6,1
73,171,6,105,0,141
1030 DATA 171,6,173,169,6,197,84,144,7
,56,229,84,160,0,240,9,14,166,6,165,84
,56,237,169,6,141
1040 DATA 172,6,173,171,6,208,8,173,17
0,6,205,172,6,144,7,169,0,240,51,76,77
,160,173,172,6,74
1050 DATA 133,212,56,173,170,6,229,212
,141,173,6,173,171,6,233,0,141,174,6,1
73,174,6,48,51,165,85
1060 DATA 172,165,6,48,15,24,105,1,133
,85,165,86,105,0,160,0,240,11,240,75,5
6,233,1,133,85,165
1070 DATA 86,233,0,133,86,173,173,6,56
,237,172,6,141,173,6,173,174,6,233,0,1
41,174,6,173,173,6
1080 DATA 24,109,170,6,141,173,6,173,1
74,6,109,171,6,141,174,6,164,84,200,17
3,166,6,16,2,136,136
```

```
1090 DATA 132,84,32,21,6,238,163,6,173
,163,6,205,172,6,208,155,96,173,171,6,
74,133,213,173,170,6
1100 DATA 106,133,212,173,172,6,56,229
,212,141,173,6,169,0,229,213,141,174,6
,173,174,6,48,31,164,84
1110 DATA 200,173,166,6,16,2,136,136,1
32,84,173,173,6,56,237,170,6,141,173,6
,173,174,6,237,171,6
1120 DATA 141,174,6,173,173,6,24,109,1
72,6,141,173,6,173,174,6,105,0,141,174
,6,165,85,172,165,6
1130 DATA 48,13,24,105,1,133,85,165,86
,105,0,160,0,240,9,56,233,1,133,85,165
,86,233,0,133,86
1140 DATA 32,21,6,238,163,6,208,3,238,
164,6,173,163,6,205,170,6,208,152,173,
164,6,205,171,6,208,144,96
1150 DATA 76,77,160,104,201,2,208,248,
104,133,86,104,133,85,104,208,239,104,
133,84,216,165,85,141,161,6
1160 DATA 165,86,141,162,6,169,0,133,2
13,168,78,162,6,110,161,6,42,78,161,6,
42,78,161,6,42,170
1170 DATA 165,84,133,212,6,212,38,213,
6,212,38,213,6,212,38,213,165,212,141,
159,6,165,213,141,160,6
1180 DATA 6,212,38,213,6,212,38,213,16
5,212,24,101,88,133,212,165,213,101,89
,133,213,165,212,24,109,159
1190 DATA 6,133,212,165,213,109,160,6,
133,213,165,212,24,109,161,6,133,212,1
65,213,109,162,6,133,213,165
1200 DATA 200,240,8,177,212,29,151,6,1
45,212,96,189,151,6,73,255,49,212,145,
212,96,128,8,32,2,64,4,16,1
```

Listing 2.

```
00010 ; FAST GRAPHICS 8 PLOT & DRAWTO
00020 ;
00030 ; Written on the SynAssembler.
00040 ;
00050 ;
00060 ; OS Equates
00070 ;
00080 SAVMSC   .EQ $58
00090 ROWCRS   .EQ $54
00100 COLCRS   .EQ $55
00110 X        .EQ COLCRS
00120 Y        .EQ ROWCRS
00130 ;
00140 ; Temporary workspace
00150 TEMP     .EQ $D4
00160 ;
00170 COLOUR   .EQ $C8 ;set by COLOR 0
00180 ;                       or 1
00190 ;
00200 ; format for draw command
00210 ;
00220 ; A=USR (START,DRAWX,DRAWY)
00230 ;
00240 ; on entry x,y = plotx,ploty
00250 ;
00260 START    PLA     ;no. of
00270          CMP #2  ;   parameters
00280          BNE ERR ;only 2 wanted
00290 ;
00300          PLA     ;Hi of draw x
00310          STA DRAW.X+1     point
00320          PLA     ;Lo of draw x
00330          STA DRAW.X       point
00340          PLA
00350          BNE ERR ;Hi byte must
00360          PLA     ;Lo byte  be 0
00370          STA DRAW.Y
00380          CLD     ;BASIC set this
00390 ;
00400 ;
00410 ; Initialise variables
00420 ;
00430 ; Note use of $40 to represent
00440 ;  a positive x or y increment
00450 ;  When this is shifted left
00460 ;  once the negative flag is set
00470 ;
00480 ;
00490 CLEAR.VAR
00500          LDA #$40  ;positive
00510          STA NEG.X ;to begin
00520          STA NEG.Y ;with
00530 ;
00540          LDA #0    ;zero loop
00550          STA I.LOOP     counter
00560          STA I.LOOP+1
00570 ;
00580 ;
00590 ; Find out the x increment
00600 ;  -> delta.x
00610 ; and its sign
00620 ;  -> neg.x
00630 ;
00640 ; DELTA.X = ABS(DRAW.X-PLOT.X)
00650 ; NEG.X   = SGN(DRAW.X-PLOT.X)
00660 ;
00670 ;
00680          LDA DRAW.X  ;Lo byte
00690          SEC     ;subtract last
00700          SBC X   ;point plotted
00710          STA DELTA.X ;save
00720          LDA DRAW.X+1
00730          SBC X+1 ;test sign of
00740 ;                MSB and set flag
00750          BPL .1
00760 ;
00770          ASL NEG.X   ;make neg
00780          EOR #$FF    ;and take
00790          STA DELTA.X+1
00800          LDA DELTA.X ;two's
00810          EOR #$FF    ;complement
00820          CLC         ;to obtain
00830          ADC #1      ;positive
00840          STA DELTA.X ;magnitude
00850          LDA DELTA.X+1
00860          ADC #0      ;maintain
00870 .1       STA DELTA.X+1
00880 ;                    Hi byte also
00890 ;
00900 ;
00910 ; Repeat the above to obtain
00920 ;  sign and magnitude of y
00930 ;  increment
00940 ; Note this is simpler than for x
00950 ;  since the maximum value of y
00960 ;  must always be less than 255
00970 ;  and hence fits into one byte
00980 ;
00990 ;
01000 ; DELTA.Y = ABS (DRAW.Y-PLOT.Y)
01010 ; NEG.Y   = SGN (DRAW.Y-PLOT.Y)
01020 ;
01030 ;
01040          LDA DRAW.Y
01050          CMP Y
01060          BLT .2
01070 ;
01080          SEC
01090          SBC Y
01100 ;
01110          LDY #0   ;unconditional
01120          BEQ .3   ;  branch
01130 ;
01140 .2       ASL NEG.Y ;set negative
01150          LDA Y    ;and calculate
01160          SEC         ;magnitude
01170          SBC DRAW.Y
01180 .3       STA DELTA.Y ;save
01190 ;
01200 ; is delta.y > delta.x
01210 ;
01220          LDA DELTA.X+1
01230          BNE .4
01240          LDA DELTA.X
01250          CMP DELTA.Y
01260          BLT X.LT.Y
01270 ;
01280 .4       LDA #0     ;a pseudo
01290          BEQ TRANSFER ;jump
01300 ;                    ;instruction
01310 ERR      JMP $A04D  ;on error
01320 ; to BASIC's warmstart address
01330 ;
01340 ;
01350 ; come here if
01360 ;  delta.y > delta.x
01370 ;
01380 ; Initialise value of error term
01390 ;
01400 ; ERROR := DELTA.X - DELTA.Y/2
01410 ;
01420 X.LT.Y
01430          LDA DELTA.Y
01440          LSR      ;divide by 2
01450          STA TEMP ;save for later
01460          SEC
```

```
01470          LDA DELTA.X
01480          SBC TEMP   ;get error
01490          STA ERROR  ;save
01500          LDA DELTA.X+1
01510          SBC #0     ;and Hi byte
01520          STA ERROR+1
01530 ;
01540 ;
01550 ; FOR I.LOOP := 1 TO DELTA.Y DO
01560 ;
01570 REPEAT.LOOP
01580 ;
01590 ;
01600 ; is ERROR > 0
01610 ;
01620          LDA ERROR+1
01630          BMI E.LT.0
01640 ;
01650 E.GT.0
01660 ;
01670 ; X := X NEG.X 1
01680 ;
01690 ; ie if neg.x is negative then
01700 ;  x=x-1
01710 ;  otherwise
01720 ;  x=x+1
01730 ;
01740          LDA X
01750          LDY NEG.X
01760          BMI SUB1    ;x negative
01770 ;
01780          CLC         ;add one
01790          ADC #1      ;to current
01800          STA X       ;point to
01810          LDA X+1     ;plot
01820          ADC #0
01830 ;
01840          LDY #0   ;unconditional
01850          BEQ STORE1 ;   jump
01860 ;
01870 ;
01880 TRANSFER BEQ X.GT.Y  ;used to
01890 ;        jump down the program
01900 ;
01910 ;
01920 SUB1     SEC
01930          SBC #1      ;subtract
01940          STA X       ;one
01950          LDA X+1
01960          SBC #0
01970 STORE1   STA X+1
01980 ;
01990 ;
02000 ;Adjust error term as follows
02010 ;
02020 ;ERROR := ERROR+DELTA.X-DELTA.Y
02030 ;
02040 ;
02050          LDA ERROR
02060          SEC
02070          SBC DELTA.Y
02080          STA ERROR
02090          LDA ERROR+1
02100          SBC #0
02110          STA ERROR+1
02120 ;
02130 ;
02140 ; ERROR less than zero
02150 ;
02160 E.LT.0
02170 ;
02180 ; ERROR := ERROR +DELTA.X
02190 ;
02200 ;
02210          LDA ERROR
02220          CLC
02230          ADC DELTA.X
02240          STA ERROR
02250          LDA ERROR+1
02260          ADC DELTA.X+1
02270          STA ERROR+1
02280 ;
02290 ;
02300 DO.Y
02310 ;
02320 ; similar to x
02330 ; Y := Y NEG.Y 1
02340 ;
02350          LDY Y
02360          INY
02370          LDA NEG.Y
02380          BPL OK1
02390 ;
02400          DEY
02410          DEY
02420 ;
02430 OK1      STY Y
02440 ;
02450 ;Now plot the point in x,y
02460 ;
02470          JSR PLOT.POINT
02480 ;
02490          INC I.LOOP  ;increment
02500          LDA I.LOOP  ;counter
02510          CMP DELTA.Y
02520          BNE REPEAT.LOOP
02530 ;
02540          RTS    ;back to BASIC
02550 ;
02560 ;
02570 ; come here if
02580 ; delta.x >=delta.y
02590 ;
02600 ;
02610 X.GT.Y
02620 ;
02630 ;initialise error term
02640 ;
02650 ; ERROR := DELTA.Y - DELTA.X/2
02660 ;
02670          LDA DELTA.X+1
02680          LSR
02690          STA TEMP+1
02700          LDA DELTA.X
02710          ROR
02720          STA TEMP
02730 ;
02740          LDA DELTA.Y
02750          SEC
02760          SBC TEMP
02770          STA ERROR
02780          LDA #0
02790          SBC TEMP+1
02800          STA ERROR+1
02810 ;
02820 ;
02830 ; FOR I.LOOP := 1 TO DELTA.X
02840 ;
02850 DO.LOOP
02860 ;
02870 ;
02880          LDA ERROR+1
02890          BMI ERROR.LT.0
02900 ;
02910 ;
02920 ; Y := Y NEG.Y 1
02930 ;
02940          LDY Y
02950          INY
02960          LDA NEG.Y
02970          BPL OK2
02980 ;
02990          DEY
03000          DEY
03010 ;
03020 OK2      STY Y
03030 ;
03040 ; ERROR :=ERROR+DELTA.Y-DELTA.X
03050 ;
03060          LDA ERROR
03070          SEC
03080          SBC DELTA.X
03090          STA ERROR
03100          LDA ERROR+1
03110          SBC DELTA.X+1
03120          STA ERROR+1
03130 ;
03140 ;
03150 ERROR.LT.0
03160 ;
03170 ; ERROR := ERROR + DELTA.Y
03180 ;
03190          LDA ERROR
03200          CLC
03210          ADC DELTA.Y
03220          STA ERROR
03230 ;
03240          LDA ERROR+1
03250          ADC #0
03260          STA ERROR+1
03270 ;
03280 ;
03290 DO.X
03300 ;
03310 ; X := X NEG.X 1
03320 ;
```

```
03330           LDA X
03340           LDY NEG.X
03350           BMI SUB4
03360 ;
03370           CLC
03380           ADC #1
03390           STA X
03400           LDA X+1
03410           ADC #0
03420 ;
03430           LDY #0
03440           BEQ STORE4
03450 ;
03460 SUB4      SEC
03470           SBC #1
03480           STA X
03490 ;
03500           LDA X+1
03510           SBC #0
03520 ;
03530 STORE4    STA X+1
03540 ;
03550           JSR PLOT.POINT
03560 ;
03570           INC I.LOOP
03580           BNE .1
03590           INC I.LOOP+1
03600 .1        LDA I.LOOP  ;two byte
03610           CMP DELTA.X ;compare
03620           BNE DO.LOOP
03630 ;
03640           LDA I.LOOP+1
03650           CMP DELTA.X+1
03660           BNE DO.LOOP
03670 ;
03680           RTS
03690 ;
03700 ;
03710 ;The plot function is stored on
03720 ; page 6 since it is non
03730 ;                   relocatable
03740 ;
03750 ; Now the plot function
03760 ;
03770 PLOT.ERROR JMP $A04D
03780 ;
03790 ;Similar protection to drawto
03800 ;                   function
03810 ;
03820 PLOT      PLA
03830           CMP #2
03840           BNE PLOT.ERROR
03850 ;
03860           PLA
03870           STA X+1
03880           PLA
03890           STA X
03900           PLA
03910           BNE PLOT.ERROR
03920           PLA
03930           STA Y
03940           CLD
03950 ;
03960 PLOT.POINT
03970 ;
03980 ; divide x by 8 saving remainder
03990 ; Do not alter current value of x
04000 ; remainder is the bit offset
04010 ; quotient is the byte offset
04020 ;
04030 ; Offsets are from top left of
04040 ;                 screen
04050 ;
04060           LDA X       ;put x
04070           STA COPY    ;into work
04080           LDA X+1     ;space
04090           STA COPY+1
04100           LDA #0
04110           STA TEMP+1
04120           TAY         ;set y reg
04130 ;
04140 ;The hi byte of x only needs to
04150 ; be shifted once since only bit
04160 ; will ever be set
04170 ;
04180 ; This saves two instructions
04190 ;
04200 ;Move bits which 'fall out' into
04210 ; the accumulator as  remainder
04220           LSR COPY+1
04230           ROR COPY
04240           ROL
04250 ;
04260           LSR COPY
04270           ROL
04280 ;
04290           LSR COPY
04300           ROL
04310 ;
04320           TAX     ;put remainder
04330           LDA Y      into x
04340           STA TEMP
04350 ;
04360 ;
04370 ; multiply y by 40 because there
04380 ; are 40 bytes per screen line
04390 ;
04400 OVER      ASL TEMP
04410           ROL TEMP+1  ;times 2
04420           ASL TEMP
04430           ROL TEMP+1  ;times 4
04440           ASL TEMP
04450           ROL TEMP+1  ;times 8
04460 ;
04470           LDA TEMP    ;save times
04480           STA EIGHT   ;8
04490           LDA TEMP+1  ;for later
04500           STA EIGHT+1
04510 ;
04520           ASL TEMP
04530           ROL TEMP+1  ;times 16
04540           ASL TEMP
04550           ROL TEMP+1  ;times 32
04560 ;
04570           LDA TEMP
04580           CLC         ;add screen
04590           ADC SAVMSC  ;address to
04600           STA TEMP
04610           LDA TEMP+1  ;temp
04620           ADC SAVMSC+1
04630           STA TEMP+1
04640 ;
04650           LDA TEMP    ;add y
04660           CLC         ;times
04670           ADC EIGHT   ;8
04680           STA TEMP    ;from
04690           LDA TEMP+1  ;earlier
04700           ADC EIGHT+1
04710           STA TEMP+1
04720 ;
04730           LDA TEMP    ;add in
04740           CLC         ;x
04750           ADC COPY    ;divided
04760           STA TEMP    ;by 8
04770           LDA TEMP+1
04780           ADC COPY+1
04790           STA TEMP+1
04800 ;
04810           LDA COLOUR  ;if COLOR 0
04820           BEQ BLANK   ;  then
04830 ;                 unplot the point
04840 ;
04850           LDA (TEMP),Y
04860           ORA PIXEL.PTR,X
04870           STA (TEMP),Y
04880           RTS         ;all done
04890 ;
04900 ;
04910 BLANK     LDA PIXEL.PTR,X get mask
04920           EOR #$FF    ;invert mask
04930           AND (TEMP),Y clear pixel
04940           STA (TEMP),Y ;to screen
04950           RTS         ;all done
04960 ;
04970 ;
04980 ;
04990 ; pixel.ptr converts  remainder
05000 ; into the mask for that pixel
05010 ;
05020 PIXEL.PTR .HS 8008200240041001
05030 ;
05040 ;
05050 ;
05060 ;General workspace
05070 ;
05080 EIGHT     .BS 2 ;8 times y
05090 COPY      .BS 2 ;a copy of x
05100 I.LOOP    .BS 2 ;loop counter
05110 NEG.X     .BS 1 ;sign of x inc.
05120 NEG.Y     .BS 1 ;sign of y inc.
05130 DRAW.X    .BS 2 ;x and y co-ords
05140 DRAW.Y    .BS 1 ;     to draw to
05150 DELTA.X   .BS 2 ;x increment
05160 DELTA.Y   .BS 1 ;y increment
05170 ERROR     .BS 2 ;two byte error
```