```
4900 RETURN
5000 REM** LOAD MCODE FROM DATA S/R **
5005 PRINTTAB(11)"*****LOADING*****"
5010 FOR I=HRSFLG TO HRSFLG+312:READ A
5020 POKE I,A:S=S+A:NEXT I
5030 READ CC:IF CC<>S THEN PRINT"CHECKSUM
ERROR":END
5040 DATA 2,0,255,255,2,2,255,255,2,18
5050 DATA 255,255,2,2,72,138,72,152,72
5060 DATA 173,0,193,240,83,169,0,133,251
5070 DATA 169,4,133,252,162,3,160,0,173
5080 DATA 2,193,145,251,136,208,251,230
5090 DATA 252,202,48,8,208,244,145,251
5100 DATA 160,231,208,238,173,1,193,240
5110 DATA 24,169,0,133,251,169,32,133
5120 DATA 252,162,32,160,0,169,0,145,251
5130 DATA 136,208,251,230,252,202,208
5140 DATA 246,173,24,208,41,240,9,8,141
5150 DATA 24,208,173,17,208,9,32,141,17
5160 DATA 208,76,125,193,173,24,208,41
5170 DATA 240,9,4,141,24,208,173,17,208
5180 DATA 41,223,141,17,208,104,168,104
5190 DATA 170,104,96,72,138,72,152,72
5200 DATA 173,4,193,141,7,193,173,3,193
5210 DATA 41,248,141,6,193,173,3,193,41
5220 DATA 7,141,8,193,173,5,193,41,7,141
5230 DATA 10,193,162,3,78,5,193,202,208
5240 DATA 250,173,5,193,141,9,193,169,0
5250 DATA 141,11,193,141,12,193,162,5
5260 DATA 173,11,193,24,109,9,193,141,11
5270 DATA 193,202,208,243,162,6,14,12
5280 DATA 193,14,11,193,144,3,238,12,193
5290 DATA 202,208,242,173,11,193,24,109
5300 DATA 6,193,141,11,193,173,12,193
5310 DATA 109,7,193,141,12,193,173,11
5320 DATA 193,24,105,0,141,11,193,173,12
5330 DATA 193,105,32,141,12,193,173,11
5340 DATA 193,24,109,10,193,141,11,193
5350 DATA 173,12,193,105,0,141,12,193
5360 DATA 173,11,193,133,251,173,12,193
5370 DATA 133,252,169,1,141,13,193,56
5380 DATA 169,7,237,8,193,170,14,13,193
5390 DATA 202,208,250,160,0,177,251,13
5400 DATA 13,193,145,251,76,125,193
5410 DATA 38698:REM*CHECKSUM*
5900 RETURN
6000 REM** MCODE ALREADY IN MEM S/R **
6100 RETURN
```

Using PLOTSUB

The demonstration BASIC program shows the various stages involved in using the machine code high resolution routines:

1) If you have an assembler, you can type in the Assembly language program, assemble it, save it as a source file, then save the object code between $C100 and $C238 under the name "PLOTSUB.HEX". Do not try to execute the subroutine at this stage, since the high resolution screen will probably immediately overwrite the assembler itself, causing it to crash.

2) To use the high resolution routines in a program, you must lower MEMTOP (see program line 220), and load the code from tape (see subroutine 4000).

3) Alternatively, you could save subroutine 5000 as a BASIC program (called "MCODELOAD", say). When you want to use it, lower MEMTOP, then load and run MCODELOAD (thus loading the machine code into memory). Type NEW, then load the program you want to run — the high resolution routines are now in memory and can be accessed by the relevant SYS instructions.

4) The last data item in subroutine 5000 is a checksum — the sum of all the preceding data. If the program stops with a "CHECKSUM ERROR" message, then you have entered the data wrongly, and must correct the mistake before proceeding.

```
;************************
;************************
;**                    **
;**  HIRES PLOTTING    **
;**                    **
;**   ROUTINE FOR      **
;**                    **
;**   COMMODORE 64     **
;**                    **
;************************
;************************
;
;
PTR     EQU $FB
MPBLO   EQU $00
MPBHI   EQU $20
SCBLO   EQU $00
SCBHI   EQU $04
SCBLK   EQU $03
SCREM   EQU $E7
MPBLK   EQU $20
        ORG $C100
;
HRSFLG DB  $00
CLRFLG DB  $00
COLOUR DB  $00
XLO    DB  $00
XHI    DB  $00
YLO    DB  $00
HBLO   DB  $00
HBHI   DB  $00
REMX   DB  $00
VBYTE  DB  $00
REMY   DB  $00
ROWLO  DB  $00
ROWHI  DB  $00
BPOS   DB  $00
;++++SWITCH TO HIRES MODE+++++
;++++AND CLEAR BIT MAP AREA+++
        PHA
        TXA
        PHA
        TYA
        PHA
        LDA HRSFLG
        BEQ RESET
;++++COLOUR SCREEN AREA+++++++
        LDA #SCBLO
        STA PTR
        LDA #SCBHI
        STA PTR+1
        LDX #SCBLK
        LDY #$00
        LDA COLOUR
AGAIN   STA (PTR),Y
        DEY
        BNE AGAIN
        INC PTR+1
        DEX
        BMI CLTEST
        BNE AGAIN
        STA (PTR),Y
        LDY #SCREM
        BNE AGAIN
CLTEST LDA CLRFLG
        BEQ HRESON
;++++CLEAR BIT MAP AREA+++++++
        LDA #MPBLO
        STA PTR
        LDA #MPBHI
        STA PTR+1
        LDX #MPBLK
        LDY #$00
        LDA #$00
NEXT    STA (PTR),Y
        DEY
        BNE NEXT
        INC PTR+1
        DEX
        BNE NEXT
;++++SET BIT MAP MODE+++++++++
HRESON LDA $D018
        AND #$F0
        ORA #$08
        STA $D018
        LDA $D011
        ORA #$20
        STA $D011
        JMP EXIT
;++++RESET TO NORMAL SCREEN+++
RESET   LDA $D018
        AND #$F0
        ORA #$04
        STA $D018
        LDA $D011
        AND #$DF
        STA $D011
;++++EXIT M/C+++++++++++++++++
EXIT    PLA
        TAY
        PLA
        TAX
        PLA
        RTS
;++++HIRES PLOT CALCULATION+++
        PHA
        TXA
        PHA
        TYA
        PHA
;++++CALCULATE HORIZ.BYTE+++++
        LDA XHI
        STA HBHI
        LDA XLO
        AND #$F8
        STA HBLO
        LDA XLO
        AND #$07
        STA REMX
;++++CALCULATE VERT.BYTE++++++
        LDA YLO
        AND #$07
        STA REMY
        LDX #$03
SHIFT   LSR YLO
        DEX
        BNE SHIFT
        LDA YLO
        STA VBYTE
;++++CALCULATE ROW++++++++++++
        LDA #$00
        STA ROWLO
        STA ROWHI
        LDX #$05
        LDA ROWLO
        CLC
        ADC VBYTE
        STA ROWLO
        DEX
        BNE FIVE
        LDX #$06
        ASL ROWHI
        ASL ROWLO
        BCC NCARRY
        INC ROWHI
        DEX
        BNE MULT
;++++ADD HORIZONTAL BYTE++++++
        LDA ROWLO
        CLC
        ADC HBLO
        STA ROWLO
        LDA ROWHI
        ADC HBHI
        STA ROWHI
;++++ADD HIRES MAP BASE+++++++
        LDA ROWLO
        CLC
        ADC #MPBLO
        STA ROWLO
        LDA ROWHI
        ADC #MPBHI
        STA ROWHI
;++++ADD REMAINDER OF YLO+++++
        LDA ROWLO
        CLC
        ADC REMY
        STA ROWLO
        LDA ROWHI
        ADC #$00
        STA ROWHI
;++++COMBINE 2 BYTES OF++++++
;++++ADDRESS ON ZERO PAGE+++++
        LDA ROWLO
        STA PTR
        LDA ROWHI
        STA PTR+1
;++++CALCULATE PIXEL POSN.++++
        LDA #$01
        STA BPOS
        SEC
        LDA #$07
        SBC REMX
        TAX
        ASL BPOS
        DEX
        BNE POWER
;++++TURN ON PIXEL++++++++++++
        LDY #$00
        LDA (PTR),Y
        ORA BPOS
        STA (PTR),Y
        JMP EXIT
```