

```

      NLS
      TTL "MTK VCX CODE 202 & 208 BOOT "
;VISICALC BOOT ROUTINE
;
;
VECT      EPZ $00
A1L       EPZ $00
A1H       EPZ $01
A2L       EPZ $02
A2H       EPZ $03
A3L       EPZ $04
A3H       EPZ $05
SLOT      EQU $BFFF
RESET     EQU $FF59
;
;
FRMPCH    EQU $76AC+$F00
ENTRY1    EQU $76AC
;
;
JSR       EQU $0020
JMP       EQU $004C
NOP       EQU $00EA
RTS       EQU $0060
LDA       EQU $00A9
;
;
      ORG $8200
      OBJ $800
;
      JMP CONTAD
;
;
VISERR     LDX #$00
VSER       LDA VSERMS,X
           BEQ VSER1
           JSR $FDED
           INX
           BNE VSER
VSER1      JMP RESET
;
;
VSERMS     HEX 8D8D878787
           INV "VERSION #S LESS THAN 200 NOT SUPPORTED !"
           HEX 8D00
;
;
CONTAD     LDA #$00
           STA $B7EB
           STA $B7EC
           STA $B7ED
           STA $B7F0
           LDA #$08
           STA $B7F1
           LDA #$01
           STA $B7F4
           JSR $3E3
           JSR $3D9
           LDY #$01
           LDA ($48),Y
           TAX
           LDA $C089,X
           LDA #CH1
           STA $84C
           LDA /CH1
           STA $84D
           JMP $801
CH1        LDA #JMP
           STA $358
           LDA #CH2
           STA $359
           LDA /CH2
           STA $35A
           JMP $301
CH2        DEY
           BPL CH2A
           JMP CH2B
CH2A       JMP $351

```

```

CH2B      LDA #JMP
           STA $FA6
           LDA #VECT
           STA $FA7
           LDA /VECT
           STA $FA8
           LDA #$2C
           STA VECT
           LDA #$81
           STA VECT+$01
           LDA #$C0
           STA VECT+$02
           LDA #JMP
           STA VECT+$03
           LDA #CH3
           STA VECT+$04
           LDA /CH3
           STA VECT+$05
           JMP ($3E)
CH3        LDA #JMP
           STA $A1
           LDA #VECT
           STA $A2
           LDA /VECT
           STA $A3
           LDA #CH4
           STA VECT+$04
           LDA /CH4
           STA VECT+$05
           JMP $6D
CH4        LDX #$00
           LDA #$6C
           CMP $6593
           BEQ CTBOOT
           LDX #$06
           CMP $6599
           BEQ CTBOOT
           JMP VISERR
CTBOOT     LDA #JMP
           STA $6593,X
           LDA #VECT
           STA $6594,X
           LDA /VECT
           STA $6595,X
           LDA #CH5
           STA VECT+$04
           LDA /CH5
           STA VECT+$05
           STX VERTYP
           JMP ($28)
VERTYP     HEX 00
CH5        LDX VERTYP
           BEQ GOOD
           CPX #$06
           BEQ GOOD
           JMP VISERR

;
;VISICALC IS LOADED, X = 0 IF
;208, 6 IF 202
;
;
;JMP AND JSR INSTRUCTIONS
;
GOOD       LDA #JMP
           STA $A6F
           STA $A14
           LDA #JSR
           STA $9D0
           STA $B02
           STA $B4A
           STA $E19
           STA $8A6
           STA $659B,X

;
;STORE THE HIGH AND LOW BYTE
;OF THE ADDRESSES TO GO TO
;
           LDA #K1

```

```
    STA $9D1
    LDA /K1
    STA $9D2
    LDA #K2
    STA $B03
    LDA /K2
    STA $B04
    LDA #K3
    STA $A70
    LDA /K3
    STA $A71
    LDA #K4
    STA $B4B
    LDA /K4
    STA $B4C
    LDA #K5
    STA $E1A
    LDA /K5
    STA $E1B
    LDA #K7
    STA $A15
    LDA /K7
    STA $A16
    LDA #K8
    STA $8A7
    LDA /K8
    STA $8A8
    LDA #K9
    STA $659C,X
    LDA /K9
    STA $659D,X
;
;PATCH OUT UPPER CASE FORCE
;
    LDA #NOP
    STA $A8C
    STA $A8D
    STA $AFE
    STA $AFF
    STA $B21
    STA $B22
    STA $A7D
    STA $A7E
    STA $AF3
    STA $AF4
    STA $A8E
    STA $A8F
    STA $B00
    STA $B01
    STA $B23
    STA $B24
    STA $A7F
    STA $A80
;
;CHANGE COLUMN WIDTH MINIMUM TO 6
;
    LDA #$06
    STA $31E2,X
;
;CHANGE RECALC INDICATOR TO "*"
;
    LDA #LDA
    STA $43AF,X
    LDA #'*'
    STA $43B0,X
    LDA #NOP
    STA $43B1,X
;
;CHANGE STORAGE TO $400, NOT $65CF
;
    LDA #BUFFER
    STA $29B0,X
    STA $29CB,X
    STA $29F6,X
    LDA /BUFFER
    STA $29B1,X
    STA $29CC,X
    STA $29F8,X
```

```
;
;ADD $28 TO FOLLOWING ADDRESSES
;
    LDY #$28
    CLC
    TYA
    ADC $6103,X
    STA $6103,X
    TYA
    ADC $4F1F,X
    STA $4F1F,X
    TYA
    ADC $39D3,X
    STA $39D3,X
    TYA
    ADC $4F26,X
    STA $4F26,X
    TYA
    ADC $43B5,X
    STA $43B5,X
    TYA
    ADC $39DD,X
    STA $39DD,X
    TYA
    ADC $4F41,X
    STA $4F41,X
    TYA
    ADC $2B8B,X
    STA $2B8B,X
    TYA
    ADC $2B8C,X
    STA $2B8C,X
    TYA
    ADC $E7C
    STA $E7C
    TYA
    ADC $B60
    STA $B60
    TYA
    ADC $A3C
    STA $A3C
    TYA
    ADC $E88
    STA $E88
    TYA
    ADC $EED
    STA $EED
;
;MOVE $400 BYTES OF PATCH CODE
;
STLOOP    LDY #$00
          LDA FRMPCH,Y
          STA ENTRY1,Y
          INY
          BNE STLOOP
STLP      LDA FRMPCH+$0100,Y
          STA ENTRY1+$0100,Y
          INY
          BNE STLP
STLP1     LDA FRMPCH+$0200,Y
          STA ENTRY1+$0200,Y
          INY
          BNE STLP1
STLP2     LDA FRMPCH+$0300,Y
          STA ENTRY1+$0300,Y
          INY
          BNE STLP2
;
;MOVE $65 BYTES
;
SCINTM    LDY #$68
          LDA $E61+$8200-$800,Y
          STA $961,Y
          DEY
          BPL SCINTM
          CLC
          TXA
;
```

```

;IF 202, THEN ADD $06 TO LOW BYTES
;OF ALL ADDRESSE THAT ARE REFERED
;TO BY PATCH ROUTINES
;
FIX      ADC V11+$01
        STA V11+$01
        TXA
        ADC V12+$01
        STA V12+$01
        TXA
        ADC V13+$01
        STA V13+$01
        TXA
        ADC SCROLB+$01
        STA SCROLB+$01
        TXA
        ADC VV1+$01
        STA VV1+$01
        TXA
        ADC VV2+$01
        STA VV2+$01
        TXA
        ADC VV3+$01
        STA VV3+$01
        TXA
        ADC VV4+$01
        STA VV4+$01
        TXA
        ADC VV5+$01
        STA VV5+$01
        TXA
        ADC K7+$01
        STA K7+$01
V12      LDA #$36
V13      STA $3148
        JMP INIT
;
;
        ORG $76AC
        OBJ $BAC
;
;
;
;VISICALC TO VIDEX VERSION II
;
;
CCPV     EPZ $75
CCPH     EPZ $76
CSPV     EPZ $77
CSPH     EPZ $78
CMPV     EPZ $7B
CMPH     EPZ $7C
WINDWD   EPZ $85
OLDHOR   EPZ $00
OLDCHR   EPZ $01
YHOLD    EPZ $02
ASAVE    EQU $4F8
BASEL    EQU $47B
BASEH    EQU $4FB
CURSOH   EQU $57B
CURSOV   EQU $5FB
N0       EQU $6F8
STARTV   EQU $6FB
POWERF   EQU $77B
FLAGS    EQU $7FB
;
;
;ENTRY #1, JSR FROM $09D0
;
;
K1       STX CURSOH
        STY CURSOV
        JSR $09D3
        LDX #$C3
        BIT $CFFF
        BIT $C300
        JMP VTAB
;

```

```
;
;ENTRY #2, JSR FROM $0B02
;
;
K2      STY YHOLD
        LDX #$C3
        LDY CURSOH
        JSR CHRPUT
        INC CURSOH
        LDY YHOLD
        INY
        RTS

;
;
;ENTRY #3, JMP FROM $0A6F
;
;
K3      PHA
        JSR K2
        JSR $0A34
        PLA
        JMP $0A76

;
;
;ENTRY #4, JSR FROM $0B4A
;
;
K4      PHA
        JSR K2
        PLA
        RTS

;
;
;VTAB IN RAM
;
;
VTAB    LDA CURSOV
        ASL
        ASL
        ADC CURSOV
        ADC STARTV
        PHA
        LSR
        LSR
        LSR
        LSR
        STA BASEH
        PLA
        ASL
        ASL
        ASL
        ASL
        STA BASEL
        RTS

;
;
;PSNCLC IN RAM
;
;
PSNCLC  CLC
        TYA
        ADC BASEL
        PHA
        LDA #$00
        ADC BASEH
        PHA
        ASL
        AND #$0C
        TAX
        LDA $C0B0,X
        PLA
        LSR
        PLA
        TAX
        RTS

;
;
;CHRPUT IN RAM
```

```
;
;
CHRPUT    EOR  #$80
          PHA
          JSR  PSNCLC
          PLA
          BCS  CHRA
          STA  $CC00,X
          RTS
CHRA      STA  $CD00,X
          RTS
;
;CURSOR MOVE  IN  RAM
;
;
CSRMOV    LDA  #$0F
          STA  $C0B0
          LDA  CURSOH
          CMP  #$50
          BCS  CSROUT
          ADC  BASEL
          STA  $C0B1
          LDA  #$0E
          STA  $C0B0
          LDA  #$00
          ADC  BASEH
          STA  $C0B1
CSROUT    RTS
;
;CURSOR POSITION, FROM $A14
;
K7        LDA  $65B6
          STA  CURSOH
          JSR  VTAB
          LDA  #$0A
          STA  $C0B0
          LDA  #$61
          STA  $C0B1
          JMP  CSRMOV
;
;CLEAR SCREEN IN RAM
;
CLSCRN    CLC
          LDY  #$00
          LDX  #$00
CLSRNA    LDA  $C0B0,Y
          LDA  #$20
CLSRNB    STA  $CC00,X
          INX
          BNE  CLSRNB
CLSRNC    STA  $CD00,X
          INX
          BNE  CLSRNC
          TYA
          ADC  #$04
          TAY
          CPY  #$10
          BNE  CLSRNA
          RTS
;
;KEYIN JSR FROM $0E19
;
;
K5        LDA  $C000
          BPL  K5
          BIT  $C010
          CMP  #$8B
          BNE  NOTK
          LDA  #$DB
NOTK       CMP  #$81
          BNE  NTSHFT
          LDA  FLAGS
          EOR  #$40
          STA  FLAGS
          JMP  K5
NTSHFT    PHA
          LDA  FLAGS
          ASL
```

[illegible]


```

        STY N0
        STA $CFFF
        JSR $C300
        LDA #$0B
        STA $C0B0
        LDA #$07
        STA $C0B1
        LDA #$0A
        STA $C0B0
        STA $C0B1
        JSR VTAB
        JSR CSRMOV
        LDX VERTYP
        LDA #INIT
        STA $5AF1,X
        LDA /INIT
        STA $5AF5,X
        LDA #$FF
        STA $9F
        LDA #$F8
        STA $9E
        LDA #$02
        STA CURSOV
V11      JMP $5A37
;
;
;ROUTINE THAT CALLS THE SCROLL
;
;
        ORG $961
        OBJ $E61
;
;
NEWSOCR  LDX #$00
        LDY #$05
        STX $65
        STY $66
VV1      LDX $65B8
VV2      LDY $65BF
        JSR VASCLC
        STX $63
        STY $64
        JSR SCROLL
        LDX #$00
        LDY #$05
        STX $63
        STY $64
VV3      LDX $65BC
VV4      LDY $65C0
        JSR VASCLC
        STX $65
        STY $66
        JSR SCROLL
VV5      LDY $65BF
        RTS
;
;SET UP FOR SCREEN CLEAR
;
K8        LDA $CFFF
        LDA $C300
        LDA $C0B0
K8A      JMP CLSCRN
;
;
;TURN OFF 80-COLUMN SCREEN
;BEFORE RE-BOOT AND EXIT
;
K9        BIT $C08A
        BIT $C0B2
        BIT $CFFF
        RTS
;
;
;SET UP ADDRESS ON SCREEN OF
;CURSOR HOR, VERT IN X,Y
;
;
VASCLC   STX CURSOH

```

```
      STY  CURSOV
VASCLA LDX  #$C3
VASCLB JSR  VTAB
      LDY  CURSOH
VASCLD JSR  PSNCLC
      LDA  #$CC
      ADC  #$00
      TAY
      RTS
ENDEND  END
```