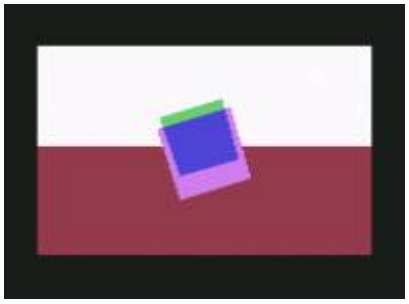
Non optimized, educational sourcecode of realtime filled vectors with calculations performed in drive.
Used assembler: TASM



Executable prg file but on the info screen in VICE enable true drive emulation 3ddrivecalc.zip

```
*= $0801
;simple basic start sys4096
.byte $0c,$08,$0a,$00,$9e,$34,$30,$39,$36,$00,$00,$00,$00


;SPECIAL FOR C&A FAN POLISH MAG
;SIMPLE DRIVECALC VECTORS
;ca-fan.com.pl

;- drive variables
ANGLEX   = $10  ;x,y,z angle
ANGLEY   = $11  ;
ANGLEZ   = $12
ZOOM     = $13  ;zoom factor
KTORYW   = $15  ;actual point in structure
DANA1    = $16  ;for calculations
;-

YPOINT   = $17   ;moment memory
ZPOINT   = $18   ;

ANGX     = $19   ;
ANGY     = $1A   ; offset for add to angle x,y,z
ANGZ     = $1B   ;
;-
SINX     = $1C   ;actual sinus value x,y,z unsigned
SINY     = $1D
SINZ     = $1E
```

```
SINXSIGN = $1F   ;sign of sine x,y,x
SINYSIGN = $20
SINZSIGN = $21

COSX     = $22    ;cosine value unsigned
COSY     = $23
COSZ     = $24

COSXSIGN = $25    ;sign of cosine
COSYSIGN = $26
COSZSIGN = $27

;-
XPRIM    = $28   ;for rotations 3d moment values
YPRIM    = $29
ZPRIM    = $2A

XBIS     = $2B
YBIS     = $2C

XLAST    = $2D
ZLAST    = $2E
;-
SIGN     = $33  ;for multiply sign of result

MNOZNA   = $34  ;data for multiply
MNOZNIK  = $35
LICZEW   = $36   ;actual point in structure


TABX2D   = $50   ;tab of x,y after perspective
TABY2D   = $68

ILEW     = $80   ;how many point in structure
BRYLA    = $81   ;data structure to calculations
;--------
         *= $1000
;--------
;here basic started
         JMP RUNPRG
;-
;send one byte to drive
SENDDRIV
         PHA
         LSR A
         LSR A
         LSR A
         LSR A
         TAX

         BIT $DD00
```

```
         BVC *-3

         SEC
X4       LDA $D012
         SBC #$32
         BCS X4    ;now only on the top and bottom border transfer
      ;  AND #$07
      ;  BEQ X4
X5
         LDA #$03
         STA $DD00
         LDA TABKON,X
         STA $DD00
         LSR A
         LSR A
         AND #$F7
         STA $DD00
         PLA
         AND #$0F
         TAX
         LDA TABKON,X
         STA $DD00
         LSR A
         LSR A
         AND #$F7
         STA $DD00
         LDA #$23
         NOP
         NOP
         NOP
         STA $DD00
         RTS
;-
;nybbles to conversion send and $DD00 save
TABKON   .BYTE $07,$87,$27,$A7,$47,$C7
         .BYTE $67,$E7
         .BYTE $17,$97,$37,$B7,$57,$D7
         .BYTE $77,$F7
;-
;get one byte from drive
POB3     BIT $DD00
         BVC *-3
         SEC
         SEI
RASTER   LDA $D012
         SBC #$32
         BCC POB3A
         AND #$07
         BEQ RASTER
POB3A
         LDA #$03
```

```
         STA $DD00
         NOP
         NOP
         NOP
         LDA #$FF
         LDX #$23
         EOR $DD00
         LSR A
         LSR A
         EOR $DD00
         LSR A
         LSR A
         EOR $DD00
         LSR A
         LSR A
         EOR $DD00
         STX $DD00
         RTS
;-
RUNPRG
;soft iinit system
         SEI
         LDA #$37
         STA $01
         JSR $FDA3
         JSR $FF5B
         LDA #$01
         STA $0286
         JSR $E544
;print the info
         LDA #<TEXT
         STA $FB
         LDA #>TEXT
         STA $FC
         LDY #$00

CNTPRINT
         LDA ($FB),Y
         BEQ EXITPRINT
         JSR $FFD2
         INY
         BNE CNTPRINT
         INC $FC
         BNE CNTPRINT

EXITPRINT

         STA $C6
         JSR $FFE4
         BEQ *-3
;wait for key
```

```
;set the vector for drive program in the ram c64
         LDA #<ADSTART
         STA MEW3+1

         LDA #0
         STA MWRT

         LDA #>ADSTART
         STA MEW3+2
         LDA $BA
         CMP #$08
         BCS NOTDRV
         LDA #$08   ;DRIVE  NR
         STA $BA
NOTDRV
;---
;error - drive not present
         LDX #3

MEW      JSR LISTEN
         BCC MEWCNT
         RTS
;memory write continue 3*32 bytes send under $0300 in drive
MEWCNT
         LDY #$05
MEW2     LDA TXMW,Y
         JSR $FFA8
         DEY
         BPL MEW2

         LDY #$20
MEW3     LDA 1000
         JSR $FFA8
         INC MWRT
         INC MEW3+1
         BNE MEW4
         INC MEW3+2
MEW4
         DEY
         BNE MEW3
         JSR $FFAE
         DEX
         BNE MEW

;-
         JSR LISTEN
;after memory write memory execute now
         LDY #$04

MEX1     LDA MEX,Y
         JSR $FFA8
```

```
         DEY
         BPL MEX1
         JSR $FFAE
         SEI
         LDA #$23
         STA $DD00

         BIT $DD00
;wait for drive program
         BVS *-3

;OK! now fast send data of all drive code
         INY
         STY $FD
         LDA #<SAVE
         STA $FB
         LDA #>SAVE
         STA $FC
         LDY #$00

SEND
         LDA ($FB),Y
         JSR SENDDRIV
         INY
         BNE SEND
         INC $FC
         INC $FD
         LDA $FD
         CMP #$05
         BNE SEND

;OK program resided in drive now
;-
         JMP DALEJ
;-

LISTEN
         LDA #$00
         STA $90
         LDA $BA
         JSR $FFB1
         LDA #$6F
         JSR $FF93
         LDA $90
         BMI NODRIVE
         CLC
         RTS
NODRIVE  SEC
         RTS
;-
TXMW
```

```
         .BYTE $20
         .BYTE 3
MWRT     .BYTE 0
         .TEXT "W-M"
;-
MEX
         .BYTE 3
         .BYTE 0
         .TEXT "E-M"
;-
SIZE1    .BYTE 44

CVECT    .BYTE 0
;Ł□Ł□Ł□Ł□Ł□Ł□Ł□
LCFACE   .BYTE 2
WALLS    .BYTE 0,1,1,2,2,3,3,0,255,1 ;TRACE OF DRAWING
         .BYTE 4,5,5,6,6,7,7,4,255,2
COLORS   .BYTE 1,2
POZWALL  .BYTE 0
NRFACE   .BYTE 0
;Ł□Ł□Ł□Ł□Ł□Ł□Ł□
DALEJ
         SEI

         LDY SIZE1  ;value point of cube 44 now
         LDA BRYLAX
         ASL A
         ADC BRYLAX
         TAX

RESIZE
         LDA BRYLAX,X
         PHP
         TYA
         PLP
         BPL NOWEB
         EOR #$FF  ;or -44
         CLC
         ADC #$01
NOWEB
         STA BRYLAX,X
         DEX
         BNE RESIZE


         JSR INITMLTCHR ;prepare multicolor chargen
         JSR SENDALL    ;send data to drive

;main loop here
LOOPTO
         SEI
```

```
CALCOS
         JSR GETROT ;get precalculated data
         JSR SENDROT;send operation type calculate again
         JSR CLRCHR;clear buffer


         LDA #$00    ;drawing faces from 0 number of face
         STA POZWALL
         STA NRFACE

LOOPCOLOR
         LDX NRFACE
         LDA COLORS,X
         STA COLOR
LOOPWALL
         LDA POZWALL
         ASL A
         TAX

         LDA WALLS,X
         BMI ENDSC

         TAY
         LDA MTABX2D,Y
         CLC

         ADC #64
         LSR A  ;MULTICOLOR
         STA MX1

         LDA MTABY2D,Y
         CLC
         ADC #64
         STA MY1

         INX
         LDA WALLS,X
         TAY

         LDA MTABX2D,Y
         CLC
         ADC #64
         LSR A  ;MULTICOLOR
         STA MX2

         LDA MTABY2D,Y
         CLC
         ADC #64
         STA MY2

         JSR DRAW  ;DRAW LINE
```

```
         INC POZWALL
         JMP LOOPWALL

ENDSC    INC POZWALL
         INC NRFACE  ;counter of faces
         LDA NRFACE
         CMP LCFACE
         BNE LOOPCOLOR

;-
;slow filled routine
         LDX #$00

         LDA #$FF

         EOR $3000,X
         STA $2000,X

         INX
         BNE *-7


         LDA #$FF

         EOR $3100,X
         STA $2100,X

         INX
         BNE *-7

         LDA #$FF

         EOR $3200,X
         STA $2200,X

         INX
         BNE *-7


         LDA #$FF

         EOR $3300,X
         STA $2300,X

         INX
         BNE *-7

         LDA #$FF
         EOR $3400,X
         STA $2400,X
```

```
         INX
         BNE *-7


         LDA #$FF
         EOR $3500,X
         STA $2500,X

         INX
         BNE *-7


         LDA #$FF
         EOR $3600,X
         STA $2600,X

         INX
         BNE *-7


         LDA #$FF

         EOR $3700,X
         STA $2700,X

         INX
         CPX #$F8
         BNE *-9

;test key for change rotation zoom etc.
         SEI
         LDA #$FD
         STA $DC00
         LDA $DC01
         ORA #$80
         CMP #$FF
         BNE TESTKEY
         LDX #$02
         STX $DC00
         CMP $DC01
         BEQ CNTN

TESTKEY
         LDA #$7F
         STA $DC00
         LDA $DC01

         JMP CHKEY

CNTN2
         LDA $DC01
```

```
          CMP $DC01
          BEQ *-3
CNTN
          JMP LOOPTO

;Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ
□Ĺ□
CHKEY
          CMP #$EF
          BNE CHKEY0
          CMP $DC01
          BEQ *-3
          LDA #$00
          STA $DC00

          LDA $DC01
          CMP $DC01
          BEQ *-3
          JMP CNTN2

CHKEY0
          CMP #$BF  ;Q
          BNE CHKEY01
          CMP $DC01
          BEQ *-3
          JSR GETROT
          LDA #$00
          JSR SENDDRIV
          LDA #$08
     ;    STA $DE00
     ;    JMP ($FFFC)
          JMP $9000

CHKEY01
          LDA #$FD
          STA $DC00
          LDA $DC01
          LDX #$FE
          STX $DC00
          CMP #$7F
          BNE CHKEY5
          LDA $DC01
          CMP #$F7   ;F8
          BNE CHKEY2
          LDA ZOMIK
          SEC
          SBC #$0A
          BCC SENDPRM
STOREZOM
          STA ZOMIK
SENDPRM
```

```
         JSR GETROT
         JSR SENDPAR
         JMP CNTN
CHKEY2
         CMP #$BF   ;F6
         BNE CHKEY3
         DEC MANGLEZ
         JMP SENDKEY

CHKEY3
         CMP #$DF   ;F4
         BNE CHKEY4
         DEC MANGLEY
         JMP SENDKEY

CHKEY4
         CMP #$EF   ;F2
         BNE CHKEY9    ;!!!
         DEC MANGLEX
         JMP SENDKEY
CHKEY5
         LDA $DC01
         CMP #$F7   ;F7
         BNE CHKEY6
         LDA ZOMIK
         CLC
         ADC #$0A
         BCC STOREZOM
         BCS SENDPRM
CHKEY6
         CMP #$BF   ;F5
         BNE CHKEY7
         INC MANGLEZ
         JMP SENDKEY

CHKEY7
         CMP #$DF   ;F3
         BNE CHKEY8
         INC MANGLEY
         JMP SENDKEY

CHKEY8
         CMP #$EF   ;F1
         BNE CHKEY9
         INC MANGLEX
         JMP SENDKEY
CHKEY9
         LDA #$FB
         STA $DC00
         LDA $DC01
         CMP #$7F  ;X
```

```
         BNE CHKEY10
         LDA MANGLEX
         EOR #$FF
         CLC
         ADC #$01
         STA MANGLEX
SENDKEY
         LDA $DC01
         CMP $DC01
         BEQ *-3
         JMP SENDPRM
CHKEY10
         CMP #$EF ;C
         BNE CHKEY10A
         LDA CVECT
         EOR #"C"
         STA CVECT
         CMP #"C"
         BNE CHKEY10B
         JSR CYBER
         JMP TEST10B
CHKEY10B
         JSR NOCYBER
TEST10B
         JMP CNTN2


CHKEY10A
         CMP #$FD   ;R
         BNE CHKEY11
         CMP $DC01
         BEQ *-3
         JSR GETROT
         LDA #"Z"
         JSR SENDDRIV
         JSR SENDPAR
         JMP CNTN
CHKEY11
         LDA #$FD
         STA $DC00
         LDA $DC01
         CMP #$EF   ;Z
         BNE CHKEY12
         LDA MANGLEZ
         EOR #$FF
         CLC
         ADC #$01
         STA MANGLEZ
         JMP SENDKEY

CHKEY12
```

```
         LDA #$F7
         STA $DC00
         LDA $DC01
         CMP #$FD   ;Y
         BNE CHKEY13
         LDA MANGLEY
         EOR #$FF
         CLC
         ADC #$01
         STA MANGLEY
         JMP SENDKEY
CHKEY13
         CMP #$EF   ;B
         BNE CHKEY14
         LDX #$02
CHK13A
         LDA MANGLEX,X
         EOR #$FF
         CLC
         ADC #$01
         STA MANGLEX,X
         DEX
         BPL CHK13A
         JMP SENDKEY

CHKEY14
         LDA #$EF
         STA $DC00
         LDA $DC01
         CMP #$F7   ;0
         BNE CHKEY16
         LDA #$00
         STA MANGLEX
         STA MANGLEY
         STA MANGLEZ
         JMP SENDKEY

CHKEY16
         JMP CNTN
;Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ□Ĺ
□Ĺ□
;--- get data after projection
GETROT
         BIT $DD00
         BVC *-3
         SEI
         SEC
         LDA $D012
         SBC #$32
         BCS *-5
         LDY #$00
```

```
GETROT2
         JSR POB3
         STA MTABX2D,Y
         JSR POB3
         STA MTABY2D,Y
         INY
         CPY BRYLAX
         BNE GETROT2
         RTS
;--- operation type rotate - calculate again
SENDROT
         LDA #"R"
         JMP SENDDRIV
;---
SENDALL
         LDY #$00
SAL2
         LDA MDATA,Y
         JSR SENDDRIV
         INY
         CPY LENOBJ  ;length of data
         BNE SAL2
         RTS
;Ĺ□Ĺ□
SENDPAR
         LDY #$00
SENDP2
         LDA PARAMS,Y
         JSR SENDDRIV
         INY
         CPY #$06
         BNE SENDP2
         RTS
;--------
LENOBJ   .BYTE 32  ;length data
MDATA    .TEXT "W" ;operation type


BRYLAX   .BYTE 8   ;8 point to calculate in structure
;--------
;point of figure x,y,z 8 times
BRYLKA
;---           X  Y  Z

         .BYTE 50,50,50
         .BYTE 50,$CE,50
         .BYTE $CE,$CE,50
         .BYTE $CE,50,50

         .BYTE 50,50,$CE
         .BYTE 50,$CE,$CE
```

```
         .BYTE $CE,$CE,$CE
         .BYTE $CE,50,$CE
;---

PARAMS   .TEXT "P" ;operation send params for drive

MANGLEX  .BYTE 3  ;value angle x,y,z
MANGLEY  .BYTE 0
MANGLEZ  .BYTE 4
ZOMIK    .BYTE $FF  ;zoom factor
         .TEXT "R" ;operation code Read

;data after projections
MTABX2D
         .BYTE $FF,$FF,$FF,$FF,$FF,$FF
         .BYTE $FF,$FF,$FF,$FF,$FF,$FF
         .BYTE $FF,$FF,$FF,$FF,$FF,$FF
         .BYTE $FF,$FF,$FF,$FF,$FF,$FF
MTABY2D
         .BYTE $FF,$FF,$FF,$FF,$FF,$FF
         .BYTE $FF,$FF,$FF,$FF,$FF,$FF
         .BYTE $FF,$FF,$FF,$FF,$FF,$FF
         .BYTE $FF,$FF,$FF,$FF,$FF,$FF
;-----------------
TBADLN   = $4000 ;TABELS ADDRES

STRTLN   = $3000  ;START OF BUFFER
ENDLN    = $3800  ;END OF BUFFER

VLINE    = $4800 ;vertical line draving
FLINE    = $4D00 ;flat line

BITS     = $E0  ;color of line

DX       = $32   ;delta x,y
DY       = $33

X1       = $2A   ;points of lines to drawing
X2       = $2B

Y1       = $2C
Y2       = $2D

VECTR1   = $FB  ;vector for read many data (pointer)
;--------
INITMLTCHR

         JSR CLRCHR    ;clear chargen
         JSR PREPLINE  ;make speedcode drawnig line
         JSR NOCYBER   ;no cyber vector presentation
```

```
;now graph init
         LDA #$18
         STA $D018

         LDA $D016
         ORA #$10
         STA $D016

         LDA #$00
         STA $D020
         LDA #$06
         STA $D021

         LDA #$05
         STA $D022

         LDA #$04
         STA $D023

         LDA #$03
         STA $D024

         LDA #$04
         STA $D025
         LDX #$00

FKL
         LDA #$09

         STA $D800,X
         STA $D8E0,X
         LDA #$0A
         STA $D9E0,X
         STA $DA00,X
         STA $DAF8,X

         INX
         BNE FKL

         RTS
;--------
;cyber vector presentation
CYBER

         LDA #$04
         STA $FC
         LDX #$00
         LDA #$04
         STA $FB
         LDA #$00
```

```
CYBER1
         STA $0400,X
         STA $0500,X
         STA $0600,X
         STA $06F8,X
         INX
         BNE CYBER1

         LDX #$02

CYBER2
         TXA

         LDY #$00
CYBER3
         STA ($FB),Y
         INY
         STA ($FB),Y
         CLC
         ADC #$10
         INY
         BCC CYBER3
         LDA $FB
         CLC
         ADC #$28
         STA $FD
         LDA $FC
         ADC #$00
         STA $FE
         CPX #14
         BEQ CYBER5
         LDY #$27
CYBER4
         LDA ($FB),Y
         STA ($FD),Y
         DEY
         BPL CYBER4
CYBER5
         LDA $FB
         CLC
         ADC #$50
         STA $FB
         LDA $FC
         ADC #$00
         STA $FC
         INX
         CPX #15
         BNE CYBER2


         RTS
```

```
;Ĺ□Ĺ□Ĺ□Ĺ□
;prepare view of chargen 128x128
NOCYBER
         LDA #$04
         STA $FC
         LDX #$00
         LDA #$AC
         STA $FB
         LDA #$00

FILED1
         STA $0400,X
         STA $0500,X
         STA $0600,X
         STA $06F8,X
         INX
         BNE FILED1

         LDX #$00

PLAN1
         TXA

         LDY #$00
PLAN2
         STA ($FB),Y
         CLC
         ADC #$10
         INY
         BCC PLAN2
         LDA $FB
         CLC
         ADC #$28
         STA $FB
         LDA $FC
         ADC #$00
         STA $FC
         INX
         CPX #$10
         BNE PLAN1

         LDA $FB
         SEC
         SBC #$28
         STA $FB

         LDA $FC
         SBC #$00
         STA $FC
         LDA #$00
         LDY #$0F
```

```
         STA ($FB),Y
         RTS
;--------
;make speedcode for drawing line
PREPLINE
         LDA #<VLINE
         STA $FB
         LDA #>VLINE
         STA $FC
         LDA #<FLINE
         STA $FD
         LDA #>FLINE
         STA $FE
         LDA #<STRTLN
         STA ADLN1
         STA ADLN1A
         LDA #>STRTLN
         STA ADLN1+1
         STA ADLN1A+1
         LDX #$FF

LOP0PRP
         LDA #<BITS
         STA CELL

LOP1PRP
         LDA #$C8
         STA LINIA1B

         LDY #$00
PREPL1
         LDA LINIA1,Y
         STA ($FB),Y
         INY
         CPY #$13
         BNE PREPL1

         LDY #$00

         LDA #$88
         STA LINIA1B

PREPL2
         LDA LINIA1,Y
         STA ($FD),Y
         INY
         CPY #$13
         BNE PREPL2

         INX
         LDA $FC
```

```
         STA TBADLN+$40,X
         LDA $FE
         STA TBADLN+$80,X

         LDA $FB
         STA TBADLN,X

        ;LDA $FB
         CLC
         ADC #$13
         STA $FB
         STA $FD
         BCC PREPL3
         INC $FC
         INC $FE
PREPL3   LDY CELL
         INY
         STY CELL
         CPY #<BITS+4
         BCC LOP1PRP

         LDA ADLN1
         CLC
         ADC #$80
         STA ADLN1
         STA ADLN1A
         BCC PREPL4
         INC ADLN1+1
         INC ADLN1A+1
PREPL4   LDA ADLN1+1
         CMP #>ENDLN
         BCC LOP0PRP

         RTS
;--------
LINIA1
         TXA
         ADC DY
         BCC LINIA1A

LINIA1B  INY
         SBC DX
         BCS LINIA1B
LINIA1A  TAX

CELL     = *+1
         LDA $E0

ADLN1    = *+1
         EOR $2000,Y
```

```
ADLN1A   = *+1
         STA $2000,Y
;------
;clear data buffer
CLRCHR
         LDX #$00
         TXA

CLRC2    STA STRTLN,X
         STA STRTLN+$0100,X
         STA STRTLN+$0200,X
         STA STRTLN+$0300,X
         STA STRTLN+$0400,X
         STA STRTLN+$0500,X
         STA STRTLN+$0600,X
         STA STRTLN+$0700,X
         INX
         BNE CLRC2

DRWEX    RTS
;---
;DRAW LINE NOW
DRAW

         LDA COLOR
         AND #$03
         STA BITS+3
         ASL A
         ASL A
         STA BITS+2
         ASL A
         ASL A
         STA BITS+1
         ASL A
         ASL A
         STA BITS

         LDA MY1
         STA Y1
         LDA MY2
         STA Y2
         LDA MX1
         STA X1
         LDA MX2
         STA X2

       ; LDA X2
         SEC
         SBC X1
         BEQ DRWEX
         BCS STOREDX
```

```
         LDX X1
         LDY X2
         STX X2
         STY X1

         LDX Y1
         LDY Y2
         STX Y2
         STY Y1
         EOR #$FF
         ADC #$01
STOREDX
         STA DX
         SEC
         LDA Y2
         SBC Y1
         LDX X1
         LDY X2
         BCS VERLINE
         EOR #$FF
         ADC #$01
         STA DY

         LDA TBADLN+$80,X
         STA ADLIN1+1

         LDA TBADLN+$80,Y
         STA VECTR1+1
RYSUJE
         LDA TBADLN,X
         STA ADLIN1
         LDA TBADLN,Y
         STA VECTR1

         LDA #$60
         LDY #$00
         STA (VECTR1),Y
         LDX #$FF
         LDY Y1

         SEI

         CLC
ADLIN1   = *+1
         JSR $1000
         LDY #$00
         LDA #$8A
         STA (VECTR1),Y
         RTS
VERLINE
         STA DY
```

```
         LDA TBADLN+$40,X
         STA ADLIN1+1

         LDA TBADLN+$40,Y
         STA VECTR1+1

         JMP RYSUJE

;---
MX1      .BYTE 0
MX2      .BYTE 63
MY1      .BYTE 0
MY2      .BYTE 127
COLOR    .BYTE 3
;---
TEXT
         .TEXT "WEGI FOR C&A FAN - DRIV"
         .TEXT "ECALC VECTORS"
         .BYTE 13,13,13
         .TEXT "F1/F2 X AXIS ROTATE +/-"
         .BYTE 13
         .TEXT "F3/F4 Y AXIS ROTATE +/-"
         .BYTE 13
         .TEXT "F5/F6 Z AXIS ROTATE +/-"
         .BYTE 13,13
         .TEXT "F7/F8 ZOOM +/-"
         .BYTE 13,13
         .TEXT "X INVERSE X ROTATION"
         .BYTE 13
         .TEXT "Y INVERSE Y ROTATION"
         .BYTE 13
         .TEXT "Z INVERSE Z ROTATION"
         .BYTE 13,13
         .TEXT "B BACK ALL ROTATION"
         .BYTE 13
         .TEXT "0 STOP ROTATION"
         .BYTE 13,13
         .TEXT "R RESET POSITION"
         .BYTE 13
         .TEXT "C CYBER VECTOR SWITCH!"
         .BYTE 13,13,13
         .TEXT "Q TO QUIT OR SPACE "
         .TEXT "KEY TO PAUSE"
         .BYTE 13,13,13
         .TEXT "MADE IN POLAND 2010.01."
         .TEXT "14"
         .BYTE 0
;--------
;---     START DRIVE CODE
;--------
SAVE
```

```
;--------
DRIVERUN = $0300
         *= DRIVERUN
         .OFFS SAVE-*
OFSETTO  = DRIVERUN-SAVE
;--------
ADSTART  = SAVE+*-DRIVERUN
;---
STARTER
         SEI
         LDA #$7A
         STA $1802
         JSR SETLINE
         ;delay loop
         JSR $F5E9
         LDX #$05
GTPRG
         JSR READBLOK
         INC ADDR
         DEX
         BNE GTPRG
         JMP RUNLDR
;---
READONE
;GET ONE BYTE FROM C64
         LDY #$FF
         .BYTE $2C
;-
READBLOK LDY #$00
;-
READBT
         LDA #$00
         STA $1800
WTR1     LDA $1800
         BNE *-3
         PHP
         LDA $1800
         ASL A
         PLP
         EOR $1800
         ASL A
         ASL A
         ASL A
         NOP
         NOP
         NOP
         EOR $1800
         ASL A
         NOP
         NOP
         NOP
```

```
         EOR $1800
ADDR     = *+2
         STA $0300,Y
         STA LASTGET
       ; NOP
         INY
         BNE WTR1
SETLINE
         LDA #$08
         STA $1800
LASTGET  = *+1
         LDA #$00
         RTS
;--------
;REWRITE ON THE SELF ! :)
;--------
;NYBBLES TO CONVERT $1800 SERIAL PORT
BIN2SER  .BYTE $0F,$07,$0D,$05,$0B,$03
         .BYTE $09,$01
         .BYTE $0E,$06,$0C,$04,$0A,$02
         .BYTE $08,$00
;--------
;SEND ONE BYTE TO C64
SENDONE  TAY
         AND #$0F
         TAX
         TYA
         LSR A
         LSR A
         LSR A
         LSR A
         TAY
         SEI
         LDA #$00
         STA $1800
         LDA BIN2SER,X
         LDX $1800
         BNE *-3
         STA $1800
         ASL A
         AND #$0A
         STA $1800
         LDA BIN2SER,Y
         STA $1800
         ASL A
         AND #$0A
         STA $1800
         JMP SETLINE
;-
; STOP ROTATES
ZEROANG
```

```
         LDA #$00
         STA ANGX
         STA ANGY
         STA ANGZ
         RTS
;-
;SLOW MULTIPLY ROUTINE
PROCKA
         BPL LP1
         EOR #$FF
         CLC
         ADC #$01

LP1      STA MNOZNA
         STY MNOZNIK

         LDA #$00
         LDX #$08
LP1A
         ROR MNOZNIK
         BCC LP2
         CLC
         ADC MNOZNA
LP2      ROR A
         DEX
         BNE LP1A


         BIT SIGN
         BPL LP3

         EOR #$FF
         CLC
         ADC #$01
LP3
         RTS

;--------
ROTATES
         LDX #$02
;--------
; COSINUS ZYX & SIGN COSINUS ZYX
;--------
FINDSINE
;---
         LDA #$00        ;SIGN +
         STA COSXSIGN,X

         LDA ANGLEX,X
         CLC
         ADC ANGX,X
```

```
         STA ANGX,X

         CMP #$40        ;COS ANGLE ZYX
         BCC FINDS3      ;SIGN +
         CMP #$C0
         BCS FINDS2      ;SIGN +
         DEC COSXSIGN,X  ;SIGN -
FINDS2
         CLC      ;ADD QUARTER PERIOD
FINDS3   ADC #$40 ;TO COS.
         AND #$7F ;MODULO HALF PERIOD
         TAY
         LDA SINE,Y
         STA COSX,X
;------
;---     SINUS ZYX
;------
         LDA ANGX,X  ;ANG IS ALSO SIGN
         AND #$7F    ;LIKE BEFORE
         TAY
         LDA SINE,Y
         STA SINX,X
;---
         DEX
         BPL FINDSINE

;--------
         INX
;--------
ROTPOINTS
         TXA
         STA LICZEW

         ASL A
         ADC LICZEW
         TAX        ;POINTNR *3
         LDY #$00

         LDA BRYLA,X
         STA XPRIM     ;XPRIM=X
         LDA BRYLA+1,X
         STA YPOINT
         LDA BRYLA+2,X
         STA ZPOINT

;--------
;-  Y PRIM
;--------
;IF YOU WANNA TEST ROTATE YOU CAN DELETED ";" CHAR
      ;  LDX ANGLEX
      ;  BNE AXISX
```

```
      ;  LDX YPOINT
      ;  STX YBIS
      ;  JMP STOREZPRIM

AXISX
         LDY SINX
                  ;IN ACC ZPOINT
         EOR ANGX
         STA SIGN
         LDA ZPOINT   ;SIN(ANGX)*Z
         JSR PROCKA
         STA DANA1

         LDY COSX
         LDA YPOINT   ;COS(ANGX)*Y
         EOR COSXSIGN

         STA SIGN
         LDA YPOINT

         JSR PROCKA
         SEC       ;COS(X)*Y - SIN(X)*Z
         SBC DANA1 ;SUBSTR.

         STA YBIS  ;Y PRIM = Y BIS

;--------
;--- Z PRIM
;--------

         LDY SINX
         LDA YPOINT

         EOR ANGX
         STA SIGN
         LDA YPOINT   ;SIN(ANGX)*Y
         JSR PROCKA
         STA DANA1

         LDY COSX
         LDA ZPOINT   ;COS(ANGX)*Z
         EOR COSXSIGN

         STA SIGN
         LDA ZPOINT

         JSR PROCKA
         CLC        ;SIN(X)*Y + COS(X)*Z
         ADC DANA1  ;SUM

STOREZPRIM
```

```
         STA ZPRIM
;--------
;--- X BIS
;--------
      ;  LDX ANGLEY
      ;  BNE AXISY
      ;  LDX XPRIM
      ;  STX XBIS
      ;  JMP STOREZLAST
AXISY
         LDY COSY
         LDA XPRIM
         EOR COSYSIGN
         STA SIGN
         LDA XPRIM    ;COS(ANGY)*X'
         JSR PROCKA
         STA DANA1

         LDY SINY
         LDA ZPRIM    ;SIN(ANGY)*Z'
         EOR ANGY
         STA SIGN
         LDA ZPRIM

         JSR PROCKA

                ;COS(Y)*X' + SIN(Y)*Z'
         CLC
         ADC DANA1  ;SUM

         STA XBIS
;--------
;--- Z BIS = Z LAST
;--------

         LDY SINY
         LDA XPRIM
         EOR ANGY
         STA SIGN
         LDA XPRIM    ;SIN(ANGY)*X'
         JSR PROCKA
         STA DANA1

         LDY COSY
         LDA ZPRIM    ;COS(ANGY)*Z'
         EOR COSYSIGN

         STA SIGN
         LDA ZPRIM

         JSR PROCKA
```

```
         SEC
                ;SIN(Y)*X' - COS(Y)*Z'

         SBC DANA1  ;SUBSTR.

STOREZLAST
               ;!!!RULE FOR Z OBSERV.
         CLC
         ADC #$80
         TAY
         LDA ZDIV,Y
         STA ZLAST    ;ZBIS = ZLAST
;--------
;--- X LAST
;--------
      ;  LDA YBIS
      ;  LDX ANGLEZ
      ;  BNE AXISZ
      ;  LDX XBIS
      ;  STX XLAST
      ;  JMP GETYLAST
AXISZ
         LDY SINZ
         LDA YBIS
         EOR ANGZ
         STA SIGN
         LDA YBIS     ;SIN(ANGZ)*Y''
         JSR PROCKA
         STA DANA1

         LDY COSZ
         LDA XBIS     ;COS(ANGZ)*X''
         EOR COSZSIGN
         STA SIGN
         LDA XBIS

         JSR PROCKA
         SEC
              ;SIN(Z)*Y'' - COS(Z)*X''

         SBC DANA1  ;SUBSTR.
         STA XLAST
;--------
;--- Y LAST
;--------
         LDY SINZ
         LDA XBIS
         EOR ANGZ
         STA SIGN
         LDA XBIS     ;SIN(ANGZ)*X
         JSR PROCKA
```

```
         STA DANA1

         LDY COSZ
         LDA YBIS     ;COS(ANGZ)*Y
         EOR COSZSIGN
         STA SIGN
         LDA YBIS

         JSR PROCKA
         CLC        ;SIN(Z)*X + COS(Z)*Y
         ADC DANA1  ;SUM

GETYLAST
        ;TAY        ;YLAST
;-------
PRSPCT
;---
         LDY ZLAST
         EOR ZLAST  ;LOSE ACC
         STA SIGN
         EOR ZLAST  ;RECALL ACC

         JSR PROCKA
         LDY ZOOM
         CPY #$FC  ;ZOOM  FACTOR =1?
         BCS STOREY2D
         STA SIGN
         TAX       ;ONLY FOR BPL/BMI
         JSR PROCKA
STOREY2D
         LDX LICZEW
         STA TABY2D,X
;----------
         LDY ZLAST
         LDA XLAST
         EOR ZLAST ;LOSE ACC
         STA SIGN
         EOR ZLAST ;RECALL ACC
PERSP4
         JSR PROCKA
         LDY ZOOM
         CPY #$FC
         BCS STOREX2D
         STA SIGN
         TAX
         JSR PROCKA

STOREX2D
         LDX LICZEW
         STA TABX2D,X
CNTROT
```

```
         INX
         CPX ILEW
         BEQ RTPSEX
         JMP ROTPOINTS
RTPSEX
         RTS
;--------
SINE
         .BYTE $00,$06,$0C,$12,$19,$1F
         .BYTE $25,$2B,$31,$38,$3E,$44
         .BYTE $4A,$50,$56,$5C,$61,$67
         .BYTE $6D,$73,$78,$7E,$83,$88
         .BYTE $8E,$93,$98,$9D,$A2,$A7
         .BYTE $AB,$B0,$B5,$B9,$BD,$C1
         .BYTE $C5,$C9,$CD,$D1,$D4,$D8
         .BYTE $DB,$DE,$E1,$E4,$E7,$EA
         .BYTE $EC,$EE,$F1,$F3,$F4,$F6
         .BYTE $F8,$F9,$FB,$FC,$FD,$FE
         .BYTE $FE,$FF,$FF,$FF,$FF,$FF
         .BYTE $FF,$FF,$FE,$FE,$FD,$FC
         .BYTE $FB,$F9,$F8,$F6,$F4,$F3
         .BYTE $F1,$EE,$EC,$EA,$E7,$E4
         .BYTE $E1,$DE,$DB,$D8,$D4,$D1
         .BYTE $CD,$C9,$C5,$C1,$BD,$B9
         .BYTE $B5,$B0,$AB,$A7,$A2,$9D
         .BYTE $98,$93,$8E,$88,$83,$7E
         .BYTE $78,$73,$6D,$67,$61,$5C
         .BYTE $56,$50,$4A,$44,$3E,$38
         .BYTE $31,$2B,$25,$1F,$19,$12
         .BYTE $0C,$06
;-----
RUNLDR
;-----
;MAIN LOOP IN DRIVE

         LDA #$00
         STA ADDR
ZERUJOBR
         JSR ZEROANG

MAINLOOP
         JSR READONE
         CMP #"Z"
         BEQ ZERUJOBR ;ZEROANG
         CMP #"P"
         BNE MLP2
         LDX #$00     ;GETPARAMS
MLP1
         JSR READONE
         STA ANGLEX,X
         INX
```

```
         CPX #$04
         BNE MLP1
         BEQ MAINLOOP

MLP2     CMP #"R"     ;ROTATE3D
         BNE MLP3
         JSR ROTATES
         LDA $1C00 ;blink
         EOR #$08
         STA $1C00
         LDX #$00
         STX LICZEW
         ;send data after projection
MLP2A
         LDA TABX2D,X
         JSR SENDONE
         LDX LICZEW
         LDA TABY2D,X
         JSR SENDONE
         INC LICZEW
         LDX LICZEW
         CPX ILEW
         BNE MLP2A
         BEQ MAINLOOP

MLP3     CMP #"W"
         BNE RESET
         JSR READONE
         STA ILEW
         ASL A
         CLC
         ADC ILEW
         STA DANA1
         LDX #$00
MLP3A
         JSR READONE
         STA BRYLA,X
         INX
         CPX DANA1
         BNE MLP3A
         BEQ MAINLOOP
RESET
         JMP ($FFFC)
;--------
;CODES:
;Z - ZEROANG
;P - GET 3 PARAMS
;R - ROTATE3D AND SEND 2D
;W - WRITE DATA OF OBJECT
;--------
ZDIV
```

```
          .BYTE $EB,$EB,$EB,$EB,$EB,$EB
          .BYTE $EB,$EB,$EB,$EB,$EB,$EA
          .BYTE $EA,$EA,$EA,$EA,$EA,$EA
          .BYTE $EA,$EA,$EA,$E9,$E9,$E9
          .BYTE $E9,$E9,$E9,$E9,$E9,$E9
          .BYTE $E8,$E8,$E8,$E8,$E8,$E8
          .BYTE $E8,$E8,$E8,$E7,$E7,$E7
          .BYTE $E7,$E7,$E7,$E7,$E7,$E6
          .BYTE $E6,$E6,$E6,$E6,$E6,$E6
          .BYTE $E5,$E5,$E5,$E5,$E5,$E5
          .BYTE $E5,$E4,$E4,$E4,$E4,$E4
          .BYTE $E4,$E3,$E3,$E3,$E3,$E3
          .BYTE $E3,$E2,$E2,$E2,$E2,$E2
          .BYTE $E2,$E1,$E1,$E1,$E1,$E1
          .BYTE $E0,$E0,$E0,$E0,$E0,$DF
          .BYTE $DF,$DF,$DF,$DE,$DE,$DE
          .BYTE $DE,$DD,$DD,$DD,$DD,$DC
          .BYTE $DC,$DC,$DC,$DB,$DB,$DB
          .BYTE $DB,$DA,$DA,$DA,$D9,$D9
          .BYTE $D9,$D9,$D8,$D8,$D8,$D7
          .BYTE $D7,$D7,$D6,$D6,$D6,$D5
          .BYTE $D5,$D4,$D4,$D4,$D3,$D3
          .BYTE $D2,$D2,$D2,$D1,$D1,$D0
          .BYTE $D0,$CF,$CF,$CE,$CE,$CD
          .BYTE $CD,$CC,$CC,$CB,$CB,$CA
          .BYTE $CA,$C9,$C8,$C8,$C7,$C6
          .BYTE $C6,$C5,$C4,$C4,$C3,$C2
          .BYTE $C1,$C1,$C0,$BF,$BE,$BD
          .BYTE $BC,$BB,$BB,$BA,$B9,$B8
          .BYTE $B7,$B5,$B4,$B3,$B2,$B1
          .BYTE $AF,$AE,$AD,$AB,$AA,$A9
          .BYTE $A7,$A5,$A4,$A2,$A0,$9E
          .BYTE $9C,$9B,$98,$96,$94,$92
          .BYTE $8F,$8D,$8A,$87,$84,$81
          .BYTE $81,$81,$81,$81,$81,$81
          .BYTE $81,$81,$81,$81,$81,$81
          .BYTE $81,$81,$81,$81,$81,$81
          .BYTE $81,$81,$81,$81,$81,$81
          .BYTE $81,$81,$81,$81,$81,$81
          .BYTE $81,$81,$81,$81,$81,$81
          .BYTE $81,$81,$81,$81,$7F,$7F
          .BYTE $7F,$7F,$7F,$7F,$7F,$7F
          .BYTE $7F,$7F,$7F,$7F
```