

Load Koala Pictures#

Assembler (BiboAssembler)#

```
00010          .LI OFF
00020 *
00030 PPOINT    .EQ $D0
00040 *
00050 SAVMSC     .EQ $58
00060 *
00070 *
00080 *
00090 *
00100 *
00110 KOALA     LDX #$10      Kanal 1
00120          JSR CLOSE     schliessen
00130          LDX #$14      Kalal 1
00140          LDA #FN       zum lesen
00150          LDY /FN       oeffnen mit
00160          JSR OPEN     Filename in FN
00170          LDX #$10      27 Bytes
00180          LDY #27       von Kanal 1
00190          LDA #$06      nach page 6 lesen
00200          JSR GETBYTES
00210          LDA $0607     Formatbyte
00220          JSR SETADD    holen und setzen
00230          JSR INITREGI  Register vorbesetzen
00240          LDY #2        Farben
00250 GCOLP1    LDA $060D,Y  in Soft-
00260          STA $2C4,Y    warere-
00270          DEY           gister
00280          BPL GCOLP1    kopieren
00290          LDA $060D+4
00300          STA $2C8
00310 -----
00320 PLOOP0     JSR GETLEN    Laenge holen
00330          LDA FFLAG     Fuellen
00340          BEQ FFILL     ja
00350 PLOOP1     JSR GETB1    Byte holen
00360          LDY YC        in Bild
00370          STA (PPOINT),Y setzen
00380          JSR ADDS      Pointer hochzaehlen
00390          LDA PLEN      Laenge
00400          BNE .1       erniedrigen
00410          DEC PLEN+1
00420 .1        DEC PLEN
00430          LDA PLEN+1
00440          BPL PLOOP1    Wiederholen wenn ungleich Null
00450          JMP PLOOP0    Weitermachen
00460 -----
00470 FFILL      JSR GETB1    Byte holen
00480          STA HREG      abspeichern
00490 PLOOP2     LDY YC        in Bildspeicher
00500          LDA HREG      kopieren
00510          STA (PPOINT),Y
00520          STA HREG
00530          JSR ADDS      naechste Bildposition setzen
00540          LDA PLEN      Laenge
```

```

00550          BNE .1          runterzaehlen
00560          DEC PLEN+1
00570 .1      DEC PLEN
00580          LDA PLEN+1
00590          BPL PLOOP2      Wiederholen bis Laenge=0
00600          JMP PLOOP0      Hauptschleife
00610 -----
00620 GETB1     LDA #0          Ein Byte
00630          TAY             holen
00640          LDX #$10        Wert steht
00650          JMP GETBYTES    im <A> Register
00660 -----
00670 INITREGI  LDA SAVMSC+1    Bildschirmadresse
00680          STA PPOINT+1    in Pointerregister
00690          LDA #0          X-Zaehler
00700          STA XC          Y-Zaehler
00710          STA YC          Zeilenzaehler
00720          STA LINEC       loeschen
00730          LDA SAVMSC
00740          STA PPOINT
00750          RTS
00760 -----
00770 ADDS      JMP $FFFF      Hier wird Sprung zur Addierungsroutine je nach Format
00780 -----
00790 SETADD    ASL             Adresse
00800          TAY             der
00810          LDA ADDT-2,Y    Addierungsroutine
00820          STA ADDS+1      oben bei
00830          LDA ADDT-1,Y    ADDS
00840          STA ADDS+2      einsetzen
00850          RTS
00860 *
00870 ADDT     .DA KAF1        Adressen der
00880          .DA KAF2        Addierungsroutinen
00890 -----
00900 GETLEN    JSR GETB1      Byte holen
00910          STA PLEN        Als LO-Byte speichern
00920          LDA #0          Fillflag
00930          STA FFLAG      und HI-Byte
00940          STA PLEN+1     loeschen
00950          LDA PLEN        Laenge negativ
00960          BPL SETF        Nein
00970          INC FFLAG      Flag auf fuellen setzen
00980          AND #$7F       Laenge-128
00990          STA PLEN
01000 SETF     BNE AJL        Ist Laenge=0? Nein
01010          JSR GETB1      16-Bit
01020          STA PLEN+1     Laenge
01030          JSR GETB1      holen
01040          STA PLEN        Laenge
01050 AJL      BNE .1          erniedrigen
01060          DEC PLEN+1
01070 .1      DEC PLEN
01080 GLOK     RTS
01090 -----
01100 KAF1     LDA PPOINT      Pointer+80
01110          CLC             (2 Zeilen)
01120          ADC #80
01130          STA PPOINT

```

```

01140          LDA PPOINT+1
01150          ADC #0
01160          STA PPOINT+1
01170          INC XC          XC=XC+1
01180          LDA XC
01190          CMP #$60        =96
01200          BNE KAF1OK     Nein
01210          LDA SAVMSC+1   Bildadresse
01220          STA PPOINT+1   in Pointer
01230          LDA SAVMSC
01240          STA PPOINT
01250          LDA #0          XC loeschen
01260          STA XC
01270          INC LINEC       Zeilenzaehler+1
01280          LDA LINEC
01290          CMP #2         =2
01300          BEQ TESTYC     Ja
01310          LDA PPOINT     Pointer+40
01320          CLC            (Eine
01330          ADC #40         Zeile
01340          STA PPOINT     versetzen)
01350          LDA PPOINT+1
01360          ADC #0
01370          STA PPOINT+1
01380          BNE KAF1OK     bedingungslos
01390 TESTYC  LDA #0          LINEC
01400          STA LINEC      loeschen
01410          INC YC         Naechste
01420          LDA YC         Spalte
01430          CMP #40        Spalte 40
01440          BEQ KAF1R      erreicht? Ja
01450 KAF1OK  RTS           zurueck zum Aufruf
01460 KAF1R   PLA           Rucksprungadresse
01470          PLA           Leschen, da Bild
01480          RTS           fertig.
01490 -----
01500 KAF2    INC YC         Naechste
01510          LDA YC         Spalte
01520          CMP #40        Spalte 40
01530          BNE KAF2OK     erreicht. Nein
01540          INC XC         Naechste
01550          LDA XC         Zeile
01560          CMP #192       192. erreicht
01570          BEQ KAF1R      Ja, Bild Fertig
01580          LDA PPOINT     Pointer
01590          CLC            +40
01600          ADC #40        Naechste
01610          STA PPOINT     Zeile
01620          BCC .1
01630          INC PPOINT+1
01640 .1      LDA #0         Spalte
01650          STA YC         loeschen
01660 KAF2OK  RTS           zurueck
01670 -----
01680 HREG     .HX 00         Platz
01690 YC      .HX 00         fuer
01700 XC      .HX 00         Hilfs-
01710 PLEN    .HX 0000       register
01720 FFLAG   .HX 00

```

```

01730 LINEC      .HX 00

00010           .LI OFF
00020           .OR $3000
00030 *
00040 *
00050 *
00060 *
00070 *
00080 *
00090 *
00100 S          JSR OPENS      Graphicbildschirm Oeffnen
00110           JSR KOALA      Bild einladen
00120           JSR GETKEY     Auf Taste Warten
00130 *
00140           LDA $E401      Editor
00150           PHA            oeffnen
00160           LDA $E400      und zum
00170           PHA            Assembler
00180           RTS            zurueckkehren
00190 -----
00200 OPENS      LDA #$80
00210           STA $6A        Graphic
00220           LDA #$F        15 ohne
00230           STA $2B        Text-
00240           LDA #0         fenster
00250           STA $2A        oeffnen
00260           LDA $E411      (Source
00270           PHA            code
00280           LDA $E410      sichern)
00290           PHA
00300           RTS
00310 -----
00320 GETKEY     LDA $E425      Auf Taste
00330           PHA            Warten
00340           LDA $E424
00350           PHA
00360           RTS
00370 -----
00380 *
00390 * Name des zu ladenden Bildes
00400 *
00410 -----
00420 FN          .DA "D:*.PIC",#$9B
00430 -----
00440 *
00450           .IN "D:IOPACK.INC"
00460           .IN "D:KOALA.INC"

```

ACTION#

```

PROC KPicLoad=*( )[$A2$10$A9$07$9D$42$03$A0$01$84$D5$A9$00$9D$48$03$9D$49$03
$20$56$E4$A4$D5$C0$08$D0$02$85$D4$C0$0E$30$07$C0$13$10$03$99
$B6$02$C8$C0$1C$30$DB$10$08$50$2E$42$2E$20$27$38$36$A9$00$85
$DB$85$DC$A5$58$85$D7$85$D9$A5$59$85$D8$85$DA$A9$00$85$D6$9D
$48$03$9D$49$03$20$56$E4$10$01$60$A8$29$80$85$DE$98$29$7F$D0
$18$A9$00$9D$48$03$9D$49$03$20$56$E4$85$D6$A9$00$9D$48$03$9D
$49$03$20$56$E4$85$D5$A9$00$9D$48$03$9D$49$03$20$56$E4$85$DD

```

\$A0\$00\$A5\$DD\$91\$D7\$38\$A5\$D5\$E9\$01\$85\$D5\$A5\$D6\$E9\$00\$85\$D6\$90
\$AA\$A5\$D4\$C9\$02\$D0\$0A\$E6\$D7\$D0\$49\$E6\$D8\$D0\$45\$90\$DB\$E6\$DC\$18
\$A5\$D7\$69\$50\$85\$D7\$90\$02\$E6\$D8\$A9\$60\$C5\$DC\$D0\$30\$A9\$00\$85\$DC
\$A5\$DB\$D0\$13\$A9\$01\$85\$DB\$18\$A5\$D9\$69\$28\$85\$D7\$A5\$DA\$69\$00\$85
\$D8\$90\$15\$A9\$00\$85\$DB\$18\$A5\$D9\$69\$01\$85\$D7\$85\$D9\$A5\$DA\$69\$00
\$85\$D8\$85\$DA\$A5\$D5\$D0\$07\$A5\$D6\$D0\$03\$18\$90\$A0\$A5\$DE\$10\$0D\$A9
\$00\$9D\$48\$03\$9D\$49\$03\$20\$56\$E4\$85\$DD\$18\$90\$9C]

MODULE ; FOR USER