INCPIC (INClude PICture)

INCPIC is a programme, which creates a .BAS file from a picture. The created programme contains the data needed to reproduce the picture, thus eliminating the need for the picture as an external resource.
The working of the programme is as follows:
Prerequisites:

1. A picture is expected to be in the ../../rfo-basic/data directory
2. The picture should be of type .PNG and contain as little colours as possible (e.g. 4 to 8 distinct colours)

A picture is read into the programme and all pixels are scanned for their colour. The resulting data are RLE coded and written to a string in byte format. Coding of the picture is either done in horizontal or in vertical direction (whichever is shortest). The data plus a decoding routine are written to a .BAS file in ../../rfo-basic/source, having the name of the picture.
One colour can be set to be transparent. As an extra feature the picture can be transformed to greyscale, optionally colourising it with one colour.
Only one picture can be used to produce a standalone .BAS file. Several generated .BAS files can be manually merged afterwards.

A working example with detailed explication.

- copy the three example pictures (ButterFly.png, ButterFly2.png, Flower.png) to your ../../rfo-basic/data directory
- Run INCPIC.bas
- Wait for the intro or skip it by tapping the screen
- Hit the screen to view a list of the picture files
- From the list tap on Flower.png; the picture will be shown in the top left corner of the screen
- Answer 'NO' to the question 'Save as greyscale ?'
- Answer 'YES' to 'Set transparent colour ?'
- Tap on the desired colour (in this case white) and tap 'OK'
- Optimise step 1 is performed (horizontal scan)
- Optimise step 2 is performed (vertical scan)
- Shortest of the steps is selected and Flower.bas is saved to ../../rfo-basic/source
- In the same way run INCPIC.bas with ButterFly.png and again with ButterFly2.png

Run the three generated .bas programmes to see the pictures are embedded in the code.....
When ready you'll be presented an enlarged version of the picture to better inspect it.

Now we can merge the three .bas programmes into one in order to have access to all pictures in one .bas programme.

Merging the three .bas programmes to create a simple scene, without resources.

Rename (or copy) Flower.bas to ButterFlies.bas and in your editor add the parts indicated in red and commented in blue to it. Also remove the part in dark red, such that your final listing looks exactly like this (everything in black is left as is):

```
! Created with INCPIC
! Jan 22, 2014
GR.OPEN 255,0,0,0,0,0
PAUSE 1000
GR.SCREEN w,h
ScaleX=720
ScaleY=w/h*ScaleX
sx=h/ScaleX
sy=w/ScaleY
GR.SCALE sx,sy
WAKELOCK 3
READ.FROM 1
GR.CLS
GR.SET.ANTIALIAS 0
GR.TEXT.SIZE 18
GR.TEXT.BOLD 1
POPUP "Creating picture from data",-300,0,0
! REMOVE this part
GOSUB MakeFlower
GR.BITMAP.DRAW ShowPic,FlowerPic,20,0
GR.RENDER
FOR i=20 TO 700 STEP 10
  j=j+10
```

```
  GR.MODIFY ShowPic,"x",i
  GR.MODIFY ShowPic,"y",j
  GR.RENDER
NEXT i
FOR i=700 TO 20 STEP -10
  j=j-10
  GR.MODIFY ShowPic,"x",i
  GR.MODIFY ShowPic,"y",j
  GR.RENDER
NEXT i
PAUSE 1000
GR.BITMAP.SCALE LargePic,FlowerPic,500/PHeight*PWidth,500,0
GR.BITMAP.DRAW ShowLargePic,LargePic,PWidth+20,0
GR.TEXT.SIZE 30
GR.COLOR 255,255,255,255,1
GR.TEXT.DRAW g,20,540,"Original file : Flower.png"
GR.TEXT.DRAW g,20,580,"Resolution : "+FORMAT$("##%",PWidth)+"
x"+FORMAT$("##%",PHeight)
IF GS=0 THEN
  T1$="Colour image"
ELSE
  T1$="Greyscale image"
ENDIF
IF TrnsP=0 THEN
  T2$="No transparent colour"
ELSE
          T2$="Transparent Colour (RGB) :"+FORMAT$("##%",rt)
+","+FORMAT$("##%",gt)+","+FORMAT$("##%",bt)
ENDIF
IF Colorise=0 THEN
  T3$=""
ELSE
   T3$="Colourised with (RGB) :"+FORMAT$("##%",rc)+","+FORMAT$
("##%",gc)+","+FORMAT$("##%",bc)
ENDIF
GR.TEXT.DRAW g,20,620,T1$
GR.TEXT.DRAW g,20,660,T2$
GR.TEXT.DRAW g,20,700,T3$
GR.RENDER
```

```
! End of part to REMOVE
! Enter this part manually (or copy it from this document)
DIM BF[20],BF2[20],nBF[20],nBF2[20]
GR.SET.STROKE 20
GR.COLOR 255,255,255,0,0
GR.RECT g,50,50,700,600
GR.SET.STROKE 0
GR.COLOR 255,192,255,255,1
GR.RECT g,60,60,690,400
GR.COLOR 255,128,255,128,1
GR.RECT g,60,400,690,590
GR.COLOR 255,255,155,0,1
GR.CIRCLE g,630,120,40
GR.RENDER
GR.TEXT.SIZE 40
GR.TEXT.DRAW PrTxt,100,500,"Creating Flower"
GR.RENDER
GOSUB MakeFlower
GR.BITMAP.DRAW ShowPic3,FlowerPic,500,250
GR.MODIFY PrTxt,"text","Creating ButterFly"
GOSUB MakeButterFly
GR.BITMAP.DRAW ShowPic,ButterFlyPic,60,100
GR.HIDE ShowPic
GOSUB MakeButterFly2
GR.BITMAP.DRAW ShowPic2,ButterFly2Pic,60,100
GR.HIDE ShowPic2
GR.MODIFY PrTxt,"text","Wow, here I am..."
FOR i=60 TO 350 STEP 10
  GR.HIDE ShowPic
  GR.SHOW ShowPic2
  GR.MODIFY ShowPic,"x",i
  GR.MODIFY ShowPic2,"x",i
  GR.RENDER
  PAUSE 50
  GR.HIDE ShowPic2
  GR.SHOW ShowPic
  GR.MODIFY ShowPic,"x",i
  GR.MODIFY ShowPic2,"x",i
  GR.RENDER
```

```
  PAUSE 50
NEXT i
GR.MODIFY PrTxt,"text","Do I see a flower ?"
FOR i=-1.54 TO -0.01 STEP 0.09
  x=FLOOR(176*COS(i)+355)
  y=FLOOR(176*SIN(i+0.03)+271)
  GR.HIDE ShowPic
  GR.SHOW ShowPic2
  GR.MODIFY ShowPic,"x",x
  GR.MODIFY ShowPic2,"x",x
  GR.MODIFY ShowPic,"y",y
  GR.MODIFY ShowPic2,"y",y
  GR.RENDER
  PAUSE 50
  GR.HIDE ShowPic2
  GR.SHOW ShowPic
  GR.MODIFY ShowPic,"x",x
  GR.MODIFY ShowPic2,"x",x
  GR.MODIFY ShowPic,"y",y
  GR.MODIFY ShowPic2,"y",y
  GR.RENDER
  PAUSE 50
NEXT i
GR.MODIFY PrTxt,"text","Mjammie, mjammie....."
FOR i=1 TO 20
  GR.BITMAP.SCALE BF[i],ButterFlyPic,45+i*2,45+i*2
  GR.BITMAP.DRAW nBF[i],BF[i],530-i,257-i
  GR.BITMAP.SCALE BF2[i],ButterFly2Pic,45+i*2,45+i*2
  GR.BITMAP.DRAW nBF2[i],BF2[i],530-i,257-i
  GR.HIDE nBF[i]
  GR.SHOW nBF2[i]
  GR.RENDER
  PAUSE 100
  GR.HIDE nBF2[i]
  GR.SHOW nBF[i]
  GR.RENDER
  PAUSE 120
NEXT i
GR.MODIFY PrTxt,"text","T H E  E N D !"
```

```
GR.RENDER
! This is the end of your added programme code
TONE 600,300,0
POPUP "Ready, touch screen to exit",-300,0,1
DO
  GR.TOUCH touched,x,y
UNTIL touched
DO
  GR.TOUCH touched,x,y
UNTIL !touched
WAKELOCK 5
GR.CLOSE
END
MakeFlower:
READ.NEXT Hor,GS,PWidth,PHeight
IF GS=0 THEN
  READ.NEXT TrnsP,rt,gt,bt
ELSE
  READ.NEXT TrnsP,rt
  gt=rt
  bt=rt
ENDIF
READ.NEXT Colorise,rc,gc,bc
GOSUB GetFlowerData
GR.BITMAP.CREATE FlowerPic,PWidth,PHeight
GR.BITMAP.DRAWINTO.START FlowerPic
GOSUB Decode
RETURN
! Copy this part from ButterFly.bas
MakeButterFly:
READ.NEXT Hor,GS,PWidth,PHeight
IF GS=0 THEN
  READ.NEXT TrnsP,rt,gt,bt
ELSE
  READ.NEXT TrnsP,rt
  gt=rt
  bt=rt
ENDIF
READ.NEXT Colorise,rc,gc,bc
```

```
GOSUB GetButterFlyData
GR.BITMAP.CREATE ButterFlyPic,PWidth,PHeight
GR.BITMAP.DRAWINTO.START ButterFlyPic
GOSUB Decode
RETURN
! Copy this part from ButterFly2.bas
MakeButterFly2:
READ.NEXT Hor,GS,PWidth,PHeight
IF GS=0 THEN
  READ.NEXT TrnsP,rt,gt,bt
ELSE
  READ.NEXT TrnsP,rt
  gt=rt
  bt=rt
ENDIF
READ.NEXT Colorise,rc,gc,bc
GOSUB GetButterFly2Data
GR.BITMAP.CREATE ButterFly2Pic,PWidth,PHeight
GR.BITMAP.DRAWINTO.START ButterFly2Pic
GOSUB Decode
RETURN
! End of merged routines to create pictures
Decode:
x1=0
y1=0
x2=0
y2=0
Counter=1
DO
  IF GS=0 THEN
    n=UCODE(MID$(PDat$,Counter,1))-128
    Counter=Counter+1
    r=UCODE(MID$(PDat$,Counter,1))-128
    C ounter=Counter+1
    g=UCODE(MID$(PDat$,Counter,1))-128
    Counter=Counter+1
    b=UCODE(MID$(PDat$,Counter,1))-128
    Counter=Counter+1
  ELSE
```

```
  n=UCODE(MID$(PDat$,Counter,1))-128
  Counter=Counter+1
  r=UCODE(MID$(PDat$,Counter,1))-128
  Counter=Counter+1
  g=r
  b=r
ENDIF
IF r=rt & g=gt & b=bt & TrnsP=1 THEN
  d=1
ELSE
  d=0
ENDIF
IF Colorise=1 THEN
  GR.COLOR 255,MAX(r,rc),MAX(g,gc),MAX(b,bc),1
ELSE
  GR.COLOR 255,r,g,b,1
ENDIF
IF Hor=1 THEN
  x2=x2+n-1
  IF x2<PWidth THEN
    IF d=0 THEN GR.LINE g,x1,y1,x2+1,y2
    x2=x2+1
    x1=x2
  ELSE
    WHILE x2>=PWidth-1
      IF d=0 THEN GR.LINE g,x1,y1,PWidth,y2
      x2=x2-PWidth
      x1=0
      y1=y1+1
      y2=y1
    REPEAT
    IF d=0 THEN GR.LINE g,x1,y1,x2+1,y2
      x2=x2+1
      x1=x2
  ENDIF
UNTIL y1>=PHeight
ELSE
  y2=y2+n-1
  IF y2<PHeight THEN
```

```
    IF d=0 THEN GR.LINE g,x1,y1,x2,y2+1
    y2=y2+1
    y1=y2
   ELSE
    WHILE y2>=PHeight-1
     IF d=0 THEN GR.LINE g,x1,y1,x2,PHeight
     y2=y2-PHeight
     y1=0
     x1=x1+1
     x2=x1
    REPEAT
    IF d=0 THEN GR.LINE g,x1,y1,x2,y2+1
     y2=y2+1
     y1=y2
   ENDIF
  UNTIL x1>=PWidth
  ENDIF
GR.BITMAP.DRAWINTO.END
GR.RENDER
RETURN
READ.DATA 0,0,92,155
READ.DATA 1,255,255,255
READ.DATA 0,0,0,0
GetFlowerData:
```
PDat$="Ö [illegible]

```
[illegible]"
RETURN
! Copy this part from ButterFly.bas
READ.DATA 1,0,45,45
READ.DATA 1,255,255,255
READ.DATA 0,0,0,0
GetButterFlyData:
PDat$="[illegible]"
RETURN
! Copy this part from ButterFly2.bas
READ.DATA 1,0,45,45
READ.DATA 1,255,255,255
READ.DATA 0,0,0,0
GetButterFly2Data:
PDat$="[illegible]
```

```
Ð±¬¡Ð±¬¥ƒƒ±¬¦ƒƒ±¬¥ƒƒ±¬ƒƒ¦ƒƒ±¬«ƒƒ±¬«ƒƒ±¬‹ƒƒ±¬«ƒƒ±¬«ƒƒ±¬ƒƒ"
RETURN
! End of merged data for pictures
```

Now run ButterFlies.bas to enjoy your newly created programme. I hope this example has made it clear how to use the INCPIC.bas programme.......
If you don't want to do all the copying yourself, you can also find the completed ButterFlies.bas file in the data directory of INCPIC.

Have fun,

Aat.