# How to GoDot

By Arndt Dettke

Colorizing Monochrome Images

Do you remember when you tried screen printing at school? Yes? How was it done… After you painted your original copy, you had to cut out stencil papers to work out the different colors of the resulting serigraphy. For each single color in the final painting you had to cut one particular stencil. Then, you had to separately print each color onto the paper, using the stencil sheets you produced before, and finally you had a wonderful piece of art!

Why not use the same technique on our Commodore to colorize black & white images we often find in clipart archives? Ok. Remember the "wax coating" we used in H2G#1 back in July? We worked with a rectangular mask then to achieve an image with two different brightness levels in it. These very masks is what we need. This time, I will cover how to create masks of any shape and how to use them as stencils.

We employ the following GoDot modules for this purpose: ldr.4Bit&Mask, mod.QuickMask, mod.DrawMask, and perhaps mod.MaskEd. Moreover, we use loaders for the appropriate b&w images, such as ldr.GIF, ldr.PCX-EGA, ldr.PFox-Select or ldr.HiBitMap. And: If everything works fine, I will also have a loader for geoPaint images finished at release time of this Digest issue. Some of the pictures treated here already were retrieved with a beta version of this new loader, exclusively written for you UCUGA Digest readers.

**This b&w image „menjou.gif" shall show us the steps we have to perform until it is colorized.**

1. We retrieve the GIF with ldr.GIF and additionally mod.DecodeGIFhir. The top frame of the film-strip contains the original monochrome GIF. We have no colors in it.

---

3. The original image must be saved as an intermediate 4Bit now, preferably as "*Undo 4Bit*" in Unit RAM (your REU) to have faster access to it. After that you install **mod.ClipWorks**, click "*Full*" and clear this area to green ("*ClrClp – green – Inside*"). There's nothing left but green in the image buffer at this time. Install **ldr.4Bit&Mask** and click "*Get 4Bit*" to retrieve the intermediate 4Bit from disk (switch "*Disk*" in this line to "*Undo*" if you used Unit RAM). "*Display*" the image. The background of the man is green now. Save the 4Bit image again as "*Undo 4Bit*" like above.

---

5. Step into mod.ClipWorks, use *ClrClp* for color light red, inside. Then enter ldr.4Bit&Mask and just click "*Get 4Bit*". Check the result after leaving the loader by "*Display*". Green background light red skin. Save this image as "*Undo 4Bit*" again. Now for the bow tie.

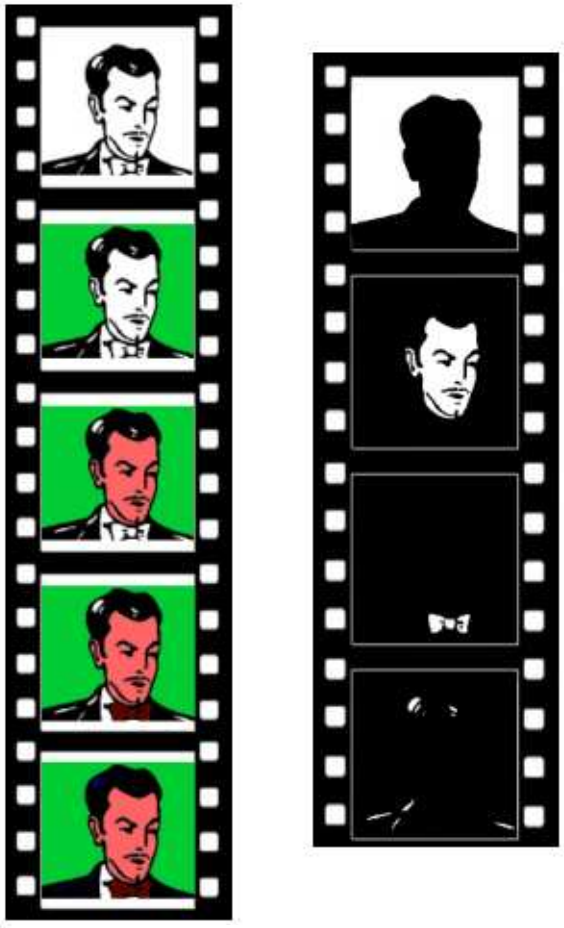
---

7. ClipWorks, *ClrClp*, red, inside. Enter ldr.4Bit&Mask and just click "*Get 4Bit*". Leave the loader, save the image like before. Finally, we treat the high lights.

---

9. Fill the 4Bit buffer with blue and get the stored image back. We're finished.

**4Bit Buffer** **Display Buffer**



Pic 1 and 2: From a GIF archive, resulting in 4 colors, plus black and white. The stencil masks on the right.

2. In **mod.QuickMask** we set color white as transparent, but then click "*Invert*" to have **all other** colors but white transparent and "*Generate*" a mask, a "stencil". You will see the face, the bow tie and the shirt of the man, like in the original image. These portions of the stencil have to be cleared away with **mod.DrawMask**, set to "*Mode: Clear*". After clicking button "*Draw Mask*" you're presented with a (sizable) green eraser brush. Edit with the left mouse button. The STOP key ends editing mode. All areas which remain white will be colorized in step 3. "*Leave*" the module.

---

4. Every step we take from now on is just a repetition of what we performed up to this stage. In mod.QuickMask we select white, invert the selection and generate the next stencil. Clear off everything which is not skin of the man and leave mod.DrawMask.

---

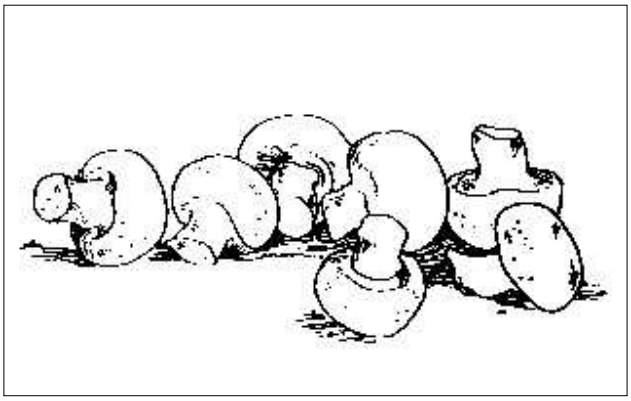
6. Relaunch mod.QuickMask, select white, invert it, and generate the stencil. There's nothing much left over: high lights in the hair and on the suit, the collar, and the bow tie. Erase everything but the bow tie with mod.DrawMask.

---

8. Generate the last mask. We just need to erase the collar and the shirt in it. The shirt is the little white band below the bow tie. Only the high lights are left over.

The monochrome images used here (and some more) are ready for download from my site at www.godot64.de. I converted most of this material to GoDot 4Bit format to give you instant access to it, as long as ldr.GeoPaint is not finally released.

Lets start. After loading a monochrome image you first display it in hires screen mode. This will show the image in full 320x200 resolution. Later, when the first colors have been applied, it is better to change screen mode to "Multi", not to run into color clash problems. The first image we discuss here is "mushrooms.4bt". It was taken from a geoPaint image called "foods.geo" using ldr.GeoPaint.



Pic 3: „mushrooms.4bt", originally from a geoPaint image

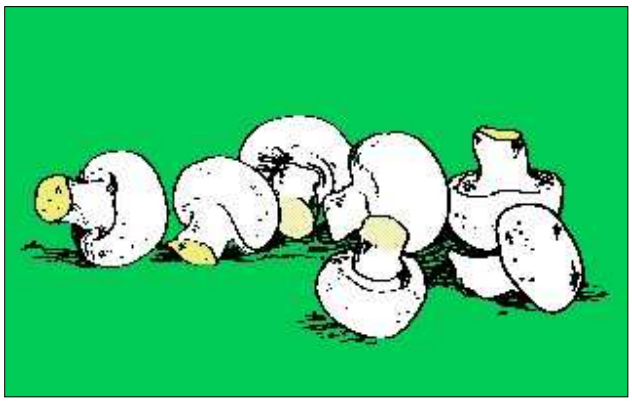
The very first you do every time when colorizing a monochrome picture is giving the background a new color. So, load the picture with **ldr.4BitGodot** and display it in hires mode. It's all white with seven mushrooms in the middle of the image. Since mushrooms naturally are white we just apply a background color (green) and only a little bit of yellow where the mushrooms have been cut off the soil.

Install **mod.QuickMask** and execute it. Select white as the color to be transparent, but immediately click "***Invert***" to have all other colors transparent (but white). "***Generate***" the mask. Have a "***View***" at it, and you will see that the mask seems to be exactly the same as the image itself. This is because the mask, too, is a monochrome image, derived from another monochrome image. The mask is stored to where normally GoDot's display buffer is located. So, please, don't "***Display***" the image until the color is applied. Displaying would destroy your mask.


Next, we have to edit the mask to produce the stencil we need. Install **mod.DrawMask** and execute it. Set the brush mode to "***Clear***" (instead of "***Draw***") and erase everything which is inside the mushrooms leaving a black shape of the food on white. "***Leave***" the module and save the 4Bit image as a temporary file, preferably as "Undo 4Bit" on unit RAM (if you're proud owner of an REU).

We will erase the image now to gain a plain green, empty 4Bit image. Install **mod.ClipWorks**, execute it, click "***Full***" and enter "***ClrClp***" to select green as the clearing color. Then engage "***Inside***", and "***Leave***" both ClrClp and ClipWorks. It is this plain color onto which the mushrooms image gets pasted, exactly where the stencil says "allowed to paste". Remember, the background area is **not** allowed and thus will stay green.

Now, install **ldr.4Bit&Mask** and enter its parameter window. You can have a look a your mask from here, too ("***View Mask***") and even invert it (if you forgot in mod.QuickMask). The important setting is "***Use Mask:***", however. If set to "***No***" you just load a 4Bit image, no matter what mask is in the buffer. If it is "***Yes***" the loaded 4Bit gets processed, and only those portions of the image where no masking bit is set will be retrieved.



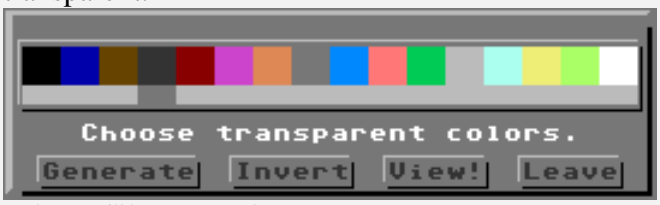
Pic 4: The final colorization, two colors, and black and white

Click the source button (where you read "Get 4Bit from ***Disk***") to "***Undo***" if you have used your REU like suggested before, and then click "***Get 4Bit***". GoDot starts loading and processing. When finished, change screen mode to "***Multi***" and "***Display***" the image. White mushrooms on green.

The yellow stem areas get colorized exactly the same way. Enter mod.QuickMask, select white, invert it, and generate a mask. Edit this mask with mod.DrawMask, set to "***Clear***", and erase everything but the stem ends of the mushrooms (five are visible). Save the 4Bit image to "Undo 4Bit", clear everything to yellow (mod.ClipWorks) and reload the undo buffer with ldr.4Bit&Mask. What you get is Pic #4.

Each additional color would also add one more pass of masking, editing, temporarily saving and color clearing until the final stage has been reached. Have a look at the examples on these pages to see how many passes were needed.

**Mod.QuickMask** – This Module opens a window in which you can choose one or more colors that will determine these as transparent when loading images with ldr.4Bit&Mask. After choosing the transparent color(s), you click on "Generate" to create a stencil layer (a hires bitmap) where every pixel of the chosen color is set to "on". This is called a "mask" within GoDot. All other pixels are "off". Have a look at the stencil with "View". You can also "Invert" the selection to gain a true negative of the stencil leaving he chosen color(s) **not** transparent.

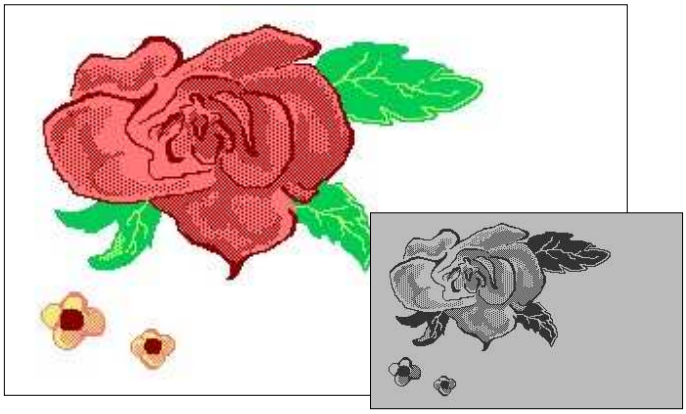



Dark gray will be transparent here.

**Mod.DrawMask** – A module to edit a mask by painting or erasing with a brush. You can change the brush size with button from 1 to 8 The drawing mode gets set with "Mode" and toggles between "Brushsize". "Draw" and "Clear". End editing with the STOP key.


**Mod.MaskEd** – Every time you find it hard to edit a mask with mod.DrawMask because you can't decently hit the very pixels you'd like to hit, use this module. It magnifies the mask bitmap by 8 and actually works like mod.PixelEdit (the only difference is that you cannot select colors). All the pictures you see here have been edited with this module, too.

**Ldr.4Bit&Mask** – Loader to retrieve 4Bit images and to process them during load. The 4Bit image won't be loaded at places where in the mask bitmap a pixel is set to "on". To "normally" load images you can switch off processing with "Use mask" set to "No".



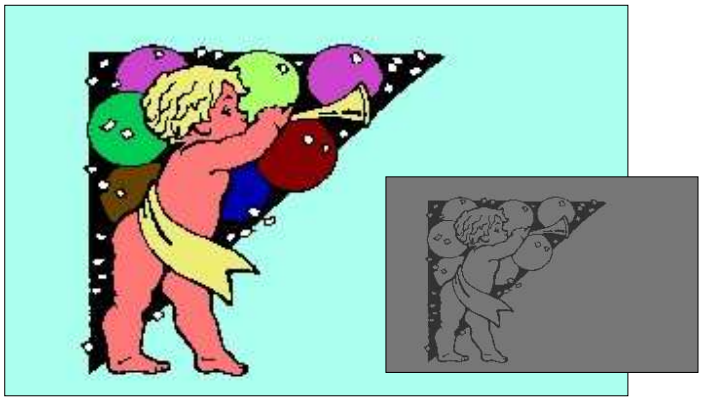
Pic 5: From a GEOS archive, contains 5 colors and white.

**Command history**
*for one color layer*

Load: 4BitGoDot (*or any loader for monochrome images*)
Load Replace "AnyMonoImage.4bt"
(Screenmode:) Hires
Display

(Repeat)
Inst: QuickMask
Execute
(Select:) white (*or the color of the monochrome background*)
Invert
Generate
(View)

Leave
Inst: DrawMask
Execute
Mode: Clear
Draw Mask
(Edit…)
<STOP> (*press the STOP key*)
Leave
Save "Undo 4Bit" (*e.g. to your REU*)
Inst: ClipWorks
Execute
Full
ClrClp
(Select:) green (*or any other applicable color*)
Inside
Leave
Accept (*or Cancel*)
Load: 4Bit&Mask
Load
Get 4Bit from: Undo (*or Disk*)
Get 4Bit
Leave
(Screenmode:) Multi
Display
(Until every color applied)

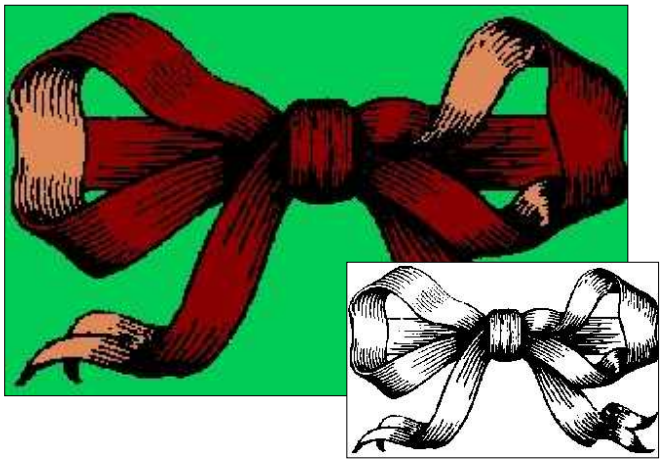


Pic 6: From a GEOS archive, contains 9 colors, black and white.

You see: the more colors you have to apply, the more effort you have to invest. Images with many, many colors in them aren't too appropriate. The upper limit is four or five colors.

For your convenience, ldr.4Bit&Mask has been updated and enhanced. You can download it from my site.

Have fun using GoDot!



Pic 7: From a GIF archive, contains 3 colors and black.